



Theory and Modeling Guide

Volume I: ADINA

ADINA 9.6

April 2020

ADINA R & D, Inc.

ADINA

Theory and Modeling Guide

Volume I: ADINA Solids & Structures

April 2020

ADINA R & D, Inc.
71 Elton Avenue
Watertown, MA 02472 USA

tel. (617) 926-5199
telefax (617) 926-0238
www.adina.com

Notices

ADINA R & D, Inc. owns both this software program system and its documentation. Both the program system and the documentation are copyrighted with all rights reserved by ADINA R & D, Inc.

The information contained in this document is subject to change without notice.

ADINA R & D, Inc. makes no warranty whatsoever, expressed or implied that the Program and its documentation including any modifications and updates are free from errors and defects. In no event shall ADINA R & D, Inc. become liable to the User or any party for any loss, including but not limited to, loss of time, money or goodwill, which may arise from the use of the Program and its documentation including any modifications and updates.

Trademarks

ADINA is a registered trademark of K.J. Bathe / ADINA R & D, Inc.

All other product names are trademarks or registered trademarks of their respective owners.

Copyright Notice

© ADINA R & D, Inc. 1987 - 2020

April 2020 Printing

Printed in the USA

Table of contents

1. Introduction	1
1.1 Objective of this manual.....	1
1.2 Supported computers and operating systems	2
1.3 Units	3
1.4 ADINA System documentation.....	5
2. Elements	9
2.1 Truss and cable elements.....	9
2.1.1 General considerations	9
2.1.2 Material models and formulations.....	11
2.1.3 Numerical integration.....	11
2.1.4 Mass matrices	12
2.1.5 Element output	13
2.1.6 Recommendations on use of elements.....	14
2.1.7 Rebar elements	14
2.2 Two-dimensional solid elements.....	17
2.2.1 General considerations	17
2.2.2 Material models and formulations.....	24
2.2.3 Numerical integration.....	26
2.2.4 Mass matrices	28
2.2.5 Element output	29
2.2.6 Recommendations on use of elements.....	36
2.3 Three-dimensional solid elements.....	40
2.3.1 General considerations	40
2.3.2 Material models and nonlinear formulations.....	48
2.3.3 Numerical integration.....	50
2.3.4 Mass matrices	53
2.3.5 Element output	53
2.3.6 Recommendations on use of elements.....	62
2.4 Beam elements	63
2.4.1 Beam geometry.....	66
2.4.2 Beam cross-section geometric properties	70
2.4.3 Beam formulation.....	72
2.4.3.1 Beam displacements	72
2.4.3.2 Displacement derivatives and strains	74
2.4.3.3 Stresses	75
2.4.3.4 Principle of virtual work.....	75
2.4.3.5 Torsional response.....	75
2.4.3.6 Strain energy for a linear elastic material.....	77

2.4.3.7	Wagner effect	79
2.4.3.8	Kinetic energy and the mass matrix	84
2.4.3.9	The warping function	85
2.4.4	Beam interpolations.....	93
2.4.4.1	Standard beam	93
2.4.4.2	Warping beam	94
2.4.4.3	Comparison of standard beam and warping beam.....	95
2.4.5	Beam element implementation	96
2.4.5.1	Linear formulation.....	97
2.4.5.2	Materially-nonlinear-only formulation.....	97
2.4.5.3	Large displacement formulation.....	97
2.4.5.4	Mass matrices	101
2.4.5.5	Elastic beam element.....	102
2.4.5.6	Elastic-plastic beam element	103
2.4.5.7	Element output	105
2.4.6	Cross-sections.....	113
2.4.6.1	General cross-section	113
2.4.6.2	Rectangular cross-section.....	114
2.4.6.3	Pipe cross-section	119
2.4.6.4	Box cross-section	122
2.4.6.5	I cross-section.....	127
2.4.6.6	U cross-section	132
2.4.6.7	L cross-section.....	137
2.4.7	Moment-curvature beam element.....	144
2.4.8	Additional features	155
2.4.9	Beam element modeling hints	160
2.4.9.1	Hints concerning the warping degrees of freedom.....	162
2.5	Iso-beam elements; axisymmetric shell elements.....	165
2.5.1	General considerations	165
2.5.2	Numerical integration.....	169
2.5.3	Linear iso-beam elements.....	171
2.5.4	Nonlinear iso-beam elements	174
2.5.5	Axisymmetric shell element.....	175
2.5.6	Element mass matrices	176
2.5.7	Element output	176
2.6	Plate elements.....	179
2.7	Shell elements.....	180
2.7.1	Basic assumptions in element formulation.....	180
2.7.2	Material models and formulations.....	188
2.7.3	Shell nodal point degrees of freedom.....	191
2.7.4	Transition elements	198

2.7.5 Numerical integration.....	200
2.7.6 Composite shell elements.....	203
2.7.7 Mass matrices.....	208
2.7.8 Anisotropic failure criteria.....	209
2.7.9 Element output.....	217
2.7.10 Selection of elements for analysis of thin and thick shells.....	223
2.7.10.1 MITC+ formulation.....	228
2.7.11 3D-shell element.....	232
2.8 Pipe elements.....	244
2.8.1 Material models and formulations.....	253
2.8.2 Pipe internal pressures.....	254
2.8.3 Numerical integration.....	257
2.8.4 Element mass matrices.....	260
2.8.5 Element output.....	261
2.8.6 Recommendations on use of elements.....	263
2.9 General and spring/damper/mass elements.....	264
2.9.1 General elements.....	264
2.9.2 Linear and nonlinear spring/damper/mass elements.....	267
2.9.3 6DOF Spring Element.....	273
2.9.4 User-supplied element.....	277
2.9.4.1 User-supplied element example.....	283
2.10 Displacement-based fluid elements.....	284
2.11 Potential-based fluid elements.....	287
2.11.1 Theory: Subsonic velocity formulation.....	289
2.11.2 Theory: Infinitesimal velocity formulation.....	296
2.11.3 Theory: Infinite fluid regions.....	298
2.11.4 Theory: Ground motion loadings.....	302
2.11.5 Theory: Static analysis.....	303
2.11.6 Theory: Frequency analysis.....	306
2.11.7 Theory: Mode superposition.....	307
2.11.8 Theory: Response spectrum, harmonic and random vibration analysis.....	309
2.11.9 Modeling: Formulation choice and potential master degree of freedom.....	311
2.11.10 Modeling: Potential degree of freedom fixities.....	312
2.11.11 Modeling: Elements.....	313
2.11.12 Modeling: Potential-interfaces.....	313
2.11.13 Modeling: Interface elements.....	315
2.11.14 Modeling: Loads.....	322
2.11.15 Modeling: Phi model completion.....	324
2.11.16 Modeling: Considerations for static analysis.....	340
2.11.17 Modeling: Considerations for dynamic analysis.....	341

2.11.18 Modeling: Considerations for frequency analysis and modal participation factor analysis.....	341
2.11.19 Modeling: Element output.....	342
2.12 Alignment elements.....	343
2.12.1 Overview	343
2.12.2 Triads.....	351
2.12.3 Notations	353
2.12.4 Translation alignment theory.....	355
2.12.5 Distance alignment theory.....	358
2.12.6 Rotation alignment theory	362
2.12.7 Triadsets and node-triadset pairs.....	368
2.12.8 Align-translation specification	370
2.12.9 Align-distance specification	371
2.12.10 Align-rotation specification.....	371
2.12.11 Element group specification.....	372
2.12.12 Element specification	372
2.12.13 Input examples	373
2.12.14 Element output	385
2.12.15 Recommendations	387
2.13 Cohesive elements.....	388
2.13.1 General considerations	388
2.13.2 Element formulation.....	388
2.13.3 Constitutive law of cohesive element.....	389
2.13.4 Numerical integration.....	392
2.13.5 Element output	392
2.14 Connector elements	393
2.14.1 Overview	393
2.14.1.1 Joint connector elements	394
2.14.1.2 Matrix connector elements	394
2.14.1.3 Multilinear matrix connector elements.....	395
2.14.1.4 Nodal triads	395
2.14.1.5 Reference axes.....	396
2.14.1.6 Undeformed configurations.....	396
2.14.2 Theory	396
2.14.2.1 Triads.....	396
2.14.2.2 Undeformed configurations and local displacements and rotations.....	400
2.14.2.3 Local forces and moments for joint type connector element.....	404
2.14.2.4 Local forces and moments for matrix type connector element.....	405
2.14.2.5 Local forces and moments for multilinear matrix type connector element	407
2.14.2.6 Consistent nodal point forces and moments.....	408

2.14.2.7 Illustrative examples.....	408
2.14.3 Modeling	428
2.14.3.1 Specification of element groups properties	428
2.14.3.2 Specification of joint type properties	429
2.14.3.3 Specification of matrix type properties	434
2.14.3.4 Specification of multilinear matrix type properties	436
2.14.3.5 Specification of element local node triads.....	438
2.14.3.6 Specification of connector element sets	441
2.14.3.7 Connector element output	441
2.14.3.8 Modeling examples	444
2.14.4 Notes and Recommendations	453
3. Material models and formulations.....	455
3.1 Stress and strain measures in ADINA	455
3.1.1 Summary	455
3.1.2 Large strain thermo-plasticity analysis with the ULH formulation.....	461
3.1.3 Large strain thermo-plasticity analysis with the ULJ formulation	466
3.1.4 Thermal strains	468
3.2 Linear elastic material models.....	471
3.2.1 Elastic-isotropic material model.....	472
3.2.2 Elastic-orthotropic material model.....	472
3.3 Nonlinear elastic material model.....	481
3.3.1 Nonlinear elastic material for truss element.....	485
3.4 Isothermal plasticity material models.....	488
3.4.1 Plastic-bilinear and plastic-multilinear material models	489
3.4.2 Mroz-bilinear material model.....	500
3.4.3 Plastic-orthotropic material model	508
3.4.4 Gurson material model	514
3.4.5 Plastic-cyclic material model	516
3.4.5.1 Fundamental concepts	517
3.4.5.2 Specification of input	536
3.4.5.3 Output variables	540
3.5 Thermo-elastic material models	541
3.6 Thermo-elasto-plasticity and creep material models.....	543
3.6.1 Thermo-elastic-plastic-creep material models.....	543
3.6.1.1 Evaluation of thermal strains.....	546
3.6.1.2 Evaluation of plastic strains	547
3.6.1.3 Evaluation of creep strains	555
3.6.1.4 Computational procedures.....	562
3.6.1.5 Shifting rule for the irradiation creep model	564
3.6.2 Thermo-plastic-cyclic material model.....	565

3.6.2.1	Fundamental concepts	566
3.6.2.2	Specification of input	574
3.6.2.3	Output variables	577
3.6.3	Evaluation of creep strains	577
3.6.4	Computational procedures.....	585
3.6.5	Shifting rule for the irradiation creep model	587
3.7	Concrete material models	588
3.7.1	Concrete material model.....	589
3.7.2	Data fitted (DF) concrete material model	607
3.7.3	Creep/Shrinkage effects	628
3.8	Rubber and foam material models.....	633
3.8.1	Isotropic hyperelastic effects.....	634
3.8.1.1	Mooney-Rivlin model	636
3.8.1.2	Ogden material model	639
3.8.1.3	Arruda-Boyce material model	641
3.8.1.4	Hyper-foam material model	643
3.8.1.5	Sussman-Bathe material model	645
3.8.1.6	Eight-chain material model	653
3.8.1.7	Curve fitting	663
3.8.2	Viscoelastic effects.....	672
3.8.2.1	Holzapfel viscoelasticity	673
3.8.2.2	Bergström-Boyce viscoelasticity.....	682
3.8.3	Mullins effect	687
3.8.4	Orthotropic effect	691
3.8.5	Thermal strain effect	695
3.8.6	TRS material.....	697
3.8.7	Temperature-dependent material.....	697
3.8.8	Rubber stability indicators.....	699
3.8.8.1	Introduction	699
3.8.8.2	Stability	700
3.8.8.3	Stability in uniaxial tension	703
3.8.8.4	Stability in pure shear.....	703
3.8.8.5	Stability in equibiaxial tension	704
3.8.8.6	Comments.....	704
3.9	Geotechnical material models	705
3.9.1	Curve description material model	705
3.9.2	Drucker-Prager material model	711
3.9.3	Cam-clay material model	716
3.9.4	Mohr-Coulomb material model.....	720
3.9.4.1	Mohr-Coulomb material model with temperature effects	726
3.10	Fabric material model with wrinkling.....	728

3.11	Viscoelastic material model	729
3.12	Porous media formulation	733
3.13	Gasket material model.....	738
3.14	Shape Memory Alloy (SMA) material model	747
3.15	The Anand material model for soldering.....	756
3.16	Piezoelectric material model	759
3.16.1	Constitutive equation.....	760
3.16.2	Orthotropic properties	762
3.16.3	Strain-form coupling matrix	763
3.16.4	Polarization direction of the piezoelectric material	764
3.16.5	Orthotropic axis system for the piezoelectric material model	765
3.16.6	Electrical loading and boundary condition.....	768
3.17	Parallel-Network Framework	769
3.17.1	Network-specific kinematics	769
3.17.2	Power expenditure, virtual velocities, and force balance	772
3.17.3	Energy dissipation	775
3.17.4	Stress update procedure.....	775
3.17.5	Three-network model	776
3.17.5.1	Rheological representation	777
3.17.5.2	Thermoelastic response	777
3.17.5.3	Viscous response	779
3.17.5.4	Constants and their interpretations	780
3.17.5.5	Additional notes	783
3.18	Cast iron material model	784
3.18.1	Fundamental concepts	785
3.18.2	Specification of input	792
3.18.3	Output variables	795
3.19	User-coded material model.....	796
3.19.1	General considerations	796
3.19.2	Input and output parameters	800
3.19.3	Stress integration procedure	804
3.19.3.1	Subincrementation.....	807
3.19.4	Coding guidelines.....	809
3.19.5	1-D truss elements	811
3.19.6	2-D solid elements.....	812
3.19.7	3-D solid elements.....	812
3.19.8	User-coded material models supplied by ADINA R&D, Inc.	813
3.19.8.1	Viscoplastic material model	814
3.19.8.2	Plasticity model	821
3.19.8.3	Thermo-elasticity and creep material model	821
3.19.8.4	Concrete material model, including creep effects	823

3.19.8.5 Viscoelastic material model	827
3.19.8.6 Ramberg-Osgood material model with mixed hardening.....	828
3.19.8.7 Nonlinear elastic material model.....	830
4. Contact conditions.....	833
4.1 Introduction	833
4.1.1 General contact.....	833
4.1.2 Rigid-target contact.....	834
4.1.3 Cat contact versions.....	834
4.1.4 Contact concepts overview.....	841
4.1.4.1 Normal contact response	841
4.1.4.2 Frictional contact.....	841
4.1.4.3 Absolute contact damping	842
4.1.4.4 Relative contact damping	842
4.1.4.5 Multiphysics	842
4.1.4.6 Contact groups.....	842
4.1.4.7 Contact surfaces	843
4.1.4.8 Contact pairs.....	844
4.1.4.9 Contact interactions	844
4.1.4.10 Contact output	845
4.1.4.11 Contact convergence criteria	845
4.1.4.12 Frequency analysis	845
4.1.4.13 Modeling recommendations	845
4.2 Normal contact	846
4.2.1 Fundamental quantities.....	846
4.2.2 Normal contact in implicit analysis.....	847
4.2.2.1 No-overlap contact	847
4.2.2.2 Compliant contact with constant compliance factor.....	850
4.2.2.3 Compliant contact with constant compliance factor and backstop..	852
4.2.2.4 Power law contact pressure function	853
4.2.2.5 XEXP contact pressure function	855
4.2.2.6 Multilinear contact pressure function	856
4.2.3 Normal contact in explicit analysis	857
4.2.3.1 Kinematic constraint algorithm	857
4.2.3.2 Penalty algorithm	858
4.2.4 Compliance energy	859
4.2.5 Specification of input	860
4.2.5.1 CGROUP and CNORMAL commands.....	860
4.2.5.2 Specification of normal contact parameters	861
4.3 Frictional contact.....	862
4.3.1 Coordinate systems.....	862

4.3.2 Slip velocity.....	862
4.3.3 Friction tractions	863
4.3.4 Friction forces	864
4.3.5 Hard-coded friction models.....	864
4.3.5.1 Coulomb friction	864
4.3.5.2 Model 1 friction.....	864
4.3.5.3 Model 2 friction.....	865
4.3.6 User-supplied friction models	867
4.3.7 Frictional contact in explicit analysis	869
4.3.7.1 Kinematic constraint method.....	869
4.3.7.2 Penalty method.....	870
4.3.8 Frictional contact dissipation energy.....	870
4.3.9 Frictional heat generation	871
4.3.10 Specification of input	871
4.3.10.1 CGROUP, CONTACTPAIR and CFRICTION commands	871
4.3.10.2 Specification of frictional contact parameters	872
4.4 Absolute contact damping	873
4.4.1 Force-based absolute contact damping.....	874
4.4.2 Traction-based absolute contact damping (not available in Cat 1 contact)	
.....	874
4.4.3 Specification of absolute contact damping	874
4.4.4 Absolute contact damping dissipation energy	875
4.4.5 Specification of input	875
4.4.5.1 CONTACT-CONTROL and CABSDAMP commands	875
4.4.5.2 Specification of absolute contact damping parameters	876
4.5 Relative contact damping	877
4.5.1 Fundamental quantities.....	877
4.5.2 Relative contact damping in implicit analysis	878
4.5.2.1 Basic relative contact damping.....	879
4.5.2.2 Power law relative contact damping.....	881
4.5.2.3 Multilinear relative contact damping.....	883
4.5.3 Relative contact damping in explicit analysis	884
4.5.4 Relative contact damping dissipation energy	884
4.5.5 Specification of input	885
4.5.5.1 CGROUP and CRELDAMP commands	885
4.5.5.2 Specification of relative contact damping parameters.....	886
4.6 Multiphysics features	886
4.6.1 Heat transfer between contact surfaces	887
4.6.1.1 Theory	887
4.6.1.2 Gap conductance options.....	889
4.6.1.3 Specification of input	892

4.6.2 Heat generation due to friction	892
4.6.2.1 Theory	892
4.6.2.2 Specification of input	895
4.7 Contact group features.....	895
4.7.1 Contact group birth/death	895
4.8 Contact surfaces	896
4.8.1 Contact surfaces for segment contact	896
4.8.1.1 2-D contact surfaces	896
4.8.1.2 3-D contact surfaces	897
4.8.1.3 Rigid and flexible surfaces	899
4.8.1.4 Contact segment normals	899
4.8.1.5 Continuous and segment normals.....	901
4.8.1.6 Single-sided contact	904
4.8.1.7 Double-sided contact	904
4.8.1.8 Contact surface extension	905
4.8.1.9 Contact segments.....	906
4.8.1.10 Contact bodies	907
4.8.1.11 Contact surface offsets	907
4.8.1.12 Contact surface depth	910
4.8.1.13 Contact slip loads	911
4.8.1.14 Automatic creation of contact surfaces	911
4.8.2 Contact surfaces for node to node contact.....	912
4.8.2.1 Analytical rigid targets for node-node contact	912
4.9 Contact pairs.....	913
4.9.1 Normal contact properties	915
4.9.2 Frictional contact properties.....	915
4.9.3 Relative contact damping properties	915
4.9.4 Multiphysics properties	916
4.9.5 Self-contact.....	916
4.9.6 Symmetric contact.....	916
4.9.7 Contact pair birth/death	916
4.10 Contact interactions and contactor elements	917
4.11 Large displacement contact	917
4.11.1 Contact search for Cat 1 contact.....	918
4.11.2 Contact search for Cat 2 contact.....	920
4.11.3 Suppression of contact oscillations with NSUPPRESS	929
4.12 Small displacement contact	930
4.13 Contactor element normal and tangential directions	932
4.14 Contactor element overlaps and gaps	933
4.15 Tied contact	933
4.16 Initial penetration features.....	935

4.16.1 Introduction	935
4.16.2 Eliminate: Immediately apply the initial contact conditions	937
4.16.3 Eliminate: Gradually apply the initial contact conditions	939
4.16.4 Gap-override: Immediately apply an initial overlap or gap	939
4.16.5 Gap-override: Gradually apply an initial overlap or gap.....	942
4.16.6 Ignore: Ignoring the initial overlap for nodes with positive overlap...	942
4.17 Contactor element relative velocities	944
4.17.1 Gap velocity	944
4.17.2 Slip velocities	944
4.17.3 Sliding distances.....	944
4.18 Contactor element equivalent areas.....	945
4.18.1 Definition of the equivalent area	945
4.18.2 Equivalent areas for higher-order contact segments.....	946
4.18.3 Equivalent areas for Cat 1 contact.....	949
4.18.4 Tension-consistent feature for Cat 1 contact	949
4.18.5 Equivalent areas for Cat 2 contact.....	949
4.18.6 Tension-consistent feature for Cat 2 contact	950
4.18.7 Equivalent areas for contact bodies.....	950
4.19 Contactor elements in implicit analysis.....	950
4.19.1 The constraint function method.....	950
4.19.1.1 The constraint function method in Cat 1 contact.....	951
4.19.1.2 The constraint function method in Cat 2 contact.....	952
4.19.1.3 Many nodes in contact.....	954
4.19.1.4 Overconstraining	955
4.19.2 Normal pressures and forces for no-overlap contact	956
4.19.2.1 Gap force vs gap relationship for Cat 1 contact	956
4.19.2.2 Gap pressure vs gap relationship for Cat 2 contact	957
4.19.3 Normal pressures and forces for compliant contact	959
4.19.3.1 Gap force vs gap relationships for Cat 1 contact.....	959
4.19.3.2 Gap pressure vs gap relationships for Cat 2 contact.....	959
4.19.4 Relative contact damping pressures and forces, Cat 2 contact.....	961
4.19.5 Frictional tractions and forces	962
4.19.5.1 Frictional constraint function, Cat 1 contact	962
4.19.5.2 Frictional constraint function, Cat 2 contact	964
4.19.6 Friction delay features	965
4.19.7 Consistent contact stiffness	966
4.19.8 Post-impact corrections	967
4.19.9 Heat transfer between contact surfaces	968
4.19.10 Heat generation due to friction	970
4.19.11 Heat transfer and heat generation with the option of using corner nodes in thermal calculations.....	971

4.20	Contact elements in explicit analysis	971
4.21	Contact element forces in explicit analysis with the penalty method	972
4.21.1	Normal forces	972
4.21.2	Relative contact damping forces	972
4.21.3	Frictional tractions and forces	973
4.22	Contact element forces in explicit analysis with the kinematic constraint method	974
4.22.1	Normal forces	974
4.22.2	Frictional forces	974
4.23	Contact output	974
4.23.1	Contact tractions and status variables	974
4.23.1.1	Status variable - contactor node states	974
4.23.1.2	Gap and overlap variables	975
4.23.1.3	Slip velocity variables	975
4.23.1.4	Segment-based contact tractions	976
4.23.1.5	Nodal-based contact tractions	977
4.23.2	Contact forces	980
4.24	Contact convergence in implicit analysis	980
4.25	Frequency analysis	985
4.25.1	Normal contact	986
4.25.2	Frictional contact	987
4.26	General modeling recommendations	988
4.26.1	Choice of contactor and target	988
4.26.2	Benefits of compliant contact, compared with no-overlap contact	989
4.26.3	General modeling hints	992
4.27	Modeling recommendations in implicit analysis	995
4.27.1	Improperly supported bodies	996
4.27.2	Contact-impact problems	999
4.27.3	Problems with many nodes in contact	1000
4.28	Modeling recommendations in explicit analysis	1001
4.28.1	Choice of explicit algorithm	1001
4.28.2	Recommendations when using the penalty algorithm	1001
4.28.3	Recommendations when using the kinematic constraint algorithm ..	1002
4.29	Rigid-target contact algorithm	1003
4.29.1	Introduction	1003
4.29.2	Basic concepts	1005
4.29.2.1	Contact surfaces	1005
4.29.2.2	Target surfaces	1009
4.29.2.3	Determination of contact between contactor and target	1009
4.29.2.4	Frictional contact	1017
4.29.2.5	Drawbeads	1019

4.29.3 Modeling considerations	1025
4.29.4 Rigid target contact reports for static and implicit dynamics	1035
4.29.5 Rigid target contact report for explicit dynamics	1038
4.29.6 Modeling hints and recommendations	1039
4.29.7 Conversion of models set up using the rigid target algorithm of ADINA	
8.3	1042
5. Boundary conditions/applied loading/constraint equations	1045
5.1 Introduction	1045
5.2 Concentrated loads	1050
5.3 Pressure and distributed loading	1053
5.3.1 Two- and three-dimensional pressure loading	1056
5.3.2 Hermitian (2-node) beam distributed loading	1059
5.3.3 Iso-beam and pipe distributed loading	1060
5.3.4 Plate/shell pressure loading	1061
5.3.5 Shell loading	1061
5.3.6 Contact surface modification of prescribed loads feature	1062
5.4 Centrifugal, rotational and mass proportional loading	1068
5.4.1 Overview	1068
5.4.2 Centrifugal loading	1071
5.4.3 Rotational loading	1074
5.4.4 Mass proportional loading	1077
5.5 Prescribed displacements, velocities & accelerations	1080
5.6 Prescribed temperatures and temperature gradients	1087
5.7 Pipe internal pressure data	1090
5.8 Electromagnetic loading	1091
5.9 Poreflow loads	1092
5.10 Phiflux loads	1092
5.11 Contact slip loads	1093
5.12 Surface tension boundary	1094
5.13 Initial load calculations	1095
5.14 User-supplied loads	1095
5.14.1 General considerations	1096
5.14.2 Usage of the user-supplied loads	1096
5.14.3 Example: Hydrodynamic forces	1099
5.15 Constraint equations	1101
5.15.1 General considerations	1101
5.15.2 Rigid links	1103
5.15.3 General rigid links	1107
5.15.3.1 Theory	1107
5.15.3.2 Modeling	1109

5.15.3.3 Notes and Recommendations	1110
5.15.4 RBE3 elements	1110
5.16 Mesh glueing	1115
5.17 Hydrostatic fluid loading	1121
6. Eigenvalue problems	1127
6.1 Linearized buckling analysis	1127
6.1.1 General considerations	1127
6.1.2 Introduction of geometric imperfections	1132
6.2 Frequency analysis	1133
6.2.1 Determinant search method	1137
6.2.2 Bathe subspace iteration method	1137
6.2.2.1 Enriched Bathe subspace iteration method	1139
6.2.3 Lanczos iteration method	1140
6.2.4 Modal stresses	1141
7. Static and implicit dynamic analysis	1145
7.1 Linear static analysis	1145
7.1.1 Sparse solver	1146
7.1.2 Iterative and multigrid solvers	1148
7.1.3 3D-iterative solver	1150
7.2 Nonlinear static analysis	1152
7.2.1 Solution of incremental nonlinear static equations	1154
7.2.2 Line search	1157
7.2.3 Low speed dynamics feature	1158
7.2.4 Automatic-Time-Stepping (ATS) method	1160
7.2.5 Total Load Application (TLA) method and Stabilized TLA (TLA-S) method	1164
7.2.6 Load-Displacement-Control (LDC) method	1167
7.2.7 Convergence criteria for equilibrium iterations	1170
7.2.8 Selection of incremental solution method	1175
7.2.9 Example	1179
7.3 Linear dynamic analysis	1185
7.3.1 Step-by-step implicit time integration	1186
7.3.2 Time history by mode superposition	1194
7.3.3 Lumped and consistent damping; Rayleigh damping	1195
7.3.4 Modal damping ratios based on strain energy proportional damping Objective	1197
7.4 Nonlinear dynamic analysis	1201
7.4.1 Step-by-step implicit time integration	1202
7.4.2 Time history by mode superposition	1204

7.4.3 Global mass matrix.....	1205
7.5 Choosing between implicit and explicit formulations.....	1207
7.6 Tracking solution progress.....	1209
7.6.1 .rto file.....	1209
7.7 Solution graphs.....	1211
8. Explicit dynamic analysis	1215
8.1 Formulation.....	1217
8.1.1 Central difference method (CDM).....	1218
8.1.2 Noh-Bathe method.....	1220
8.1.3 Mass matrix.....	1222
8.1.4 Damping – central difference method.....	1223
8.1.5 Damping – Noh-Bathe method.....	1223
8.2 Stability.....	1224
8.3 Time step management.....	1227
8.4 Tracking solution progress.....	1229
9. Frequency domain analysis	1231
9.1 Response spectrum analysis.....	1231
9.2 Fourier analysis.....	1242
9.3 Harmonic vibration analysis.....	1245
9.4 Random vibration analysis.....	1252
9.5 SDOF system response.....	1258
10. Fracture mechanics	1271
10.1 Overview.....	1271
10.2 The energy release rate.....	1274
10.3 Stress intensity factors.....	1275
10.4 Calculation of the J-integral using the line contour integration method..	1279
10.5 Calculation of the J-integral using the virtual crack extension method ..	1281
10.6 Calculation of stress intensity factors using the virtual crack extension method.....	1287
10.6.1 Overview of theory.....	1287
10.6.2 Calculation of stress intensity factors.....	1294
10.6.3 Discussion.....	1295
10.7 Fracture control parameters.....	1296
10.8 2-D crack geometry and meshing.....	1297
10.8.1 Overview.....	1297
10.8.2 Meshing recommendations.....	1300
10.8.3 Definition of the crack propagation line, no nodes on the line of self- similar crack advance	1303

10.8.4 Definition of the crack propagation line, nodes on the line of self-similar crack advance	1303
10.9 Definition and use of line contours for 2-D cracks	1304
10.9.1 Definition of line contours	1304
10.9.2 Output.....	1307
10.10 Definition and use of virtual shifts for 2-D cracks	1307
10.10.1 Domain of the virtual shift	1309
10.10.2 Virtual shift vector.....	1312
10.10.3 Output.....	1313
10.11 Crack propagation for 2-D cracks	1313
10.12 3-D crack geometry and meshing.....	1323
10.12.1 Overview	1323
10.12.2 Meshing recommendations.....	1324
10.13 3-D crack geometry and meshing using CRACK-M.....	1327
10.14 NVS method for virtual crack extension for 3-D cracks	1338
10.14.1 Overview	1338
10.14.2 Definition of the crack propagation surface, CRACK-M features are not used.....	1347
10.14.3 Definition of the virtual shifts, CRACK-M features are not used..	1349
10.14.4 Definitions of the virtual shifts when CRACK-M features are used..	1353
10.14.5 Fracture mechanics output.....	1355
10.15 SVS method for virtual crack extension for 3-D cracks.....	1356
10.15.1 Overview	1356
10.15.2 Definitions of the virtual shifts.....	1357
10.15.3 Calculation of J-integral and stress intensity factors	1366
10.15.4 Calculation of G using numerical integration	1367
10.15.5 Definitions of the cracks.....	1373
10.15.6 Virtual shift numbers assigned to the virtual shifts	1376
10.15.7 Modeling hints.....	1378
10.15.8 Fracture mechanics output.....	1380
11. Additional capabilities	1385
11.1 Substructuring	1385
11.1.1 Substructuring in linear analysis	1385
11.1.2 Substructuring in nonlinear analysis	1387
11.1.3 Substructuring restrictions.....	1388
11.1.4 Additional substructuring modeling techniques.....	1389
11.2 Cyclic symmetry analysis.....	1397
11.3 Reactions calculation.....	1403
11.4 Element birth and death feature.....	1404
11.5 Element "death upon rupture"	1412

11.6 Initial conditions.....	1413
11.6.1 Initial displacements, velocities and accelerations.....	1413
11.6.2 Initial temperatures and temperature gradients	1414
11.6.3 Initial pipe internal pressures	1415
11.6.4 Initial strains.....	1415
11.6.5 Initial stresses	1423
11.7 Energy calculation.....	1427
11.7.1 Energy printout.....	1427
11.7.2 Strain energy densities.....	1435
11.8 Element group inertial properties.....	1437
11.9 Restart.....	1438
11.10 Parallel processing.....	1443
11.11 Usage of memory and disk storage	1444
11.12 Miscellaneous.....	1445
11.13 Remeshing options	1447
11.14 Analysis zooming.....	1449
11.15 Stiffness stabilization	1451
11.16 Geometric imperfections	1454
11.17 Bolt feature	1454
11.17.1 Beam-bolt element	1457
11.17.2 3D-bolt element group.....	1461
11.17.3 Usage of bolt loadings.....	1465
11.17.4 Usage of beam-bolts	1469
11.17.5 Usage of 3D-bolts.....	1470
11.17.6 Modeling issues.....	1477
11.17.7 Post-processing variables	1477
11.18 Component mode synthesis (CMS) method.....	1478
12. Heat transfer capabilities in the ADINA program.....	1487
12.1 Analysis options	1488
12.2 Modeling	1490
12.3 Soil consolidation analysis	1493
12.4 Piezoelectric analysis	1496
12.5 Thermal coupling between solids and fluids	1498
13. Postprocessing considerations	1501
13.1 Calculation of results within the AUI.....	1501
13.1.1 What results are available.....	1507
13.1.2 Where results can be obtained.....	1544
13.1.3 How results are evaluated (including smoothing)	1555
13.1.4 When (for which load step, etc.) are results available.....	1559

13.2 Evaluation of meshes used	1565
Index	1567

1. Introduction

1.1 Objective of this manual

The objective of this manual is to give a concise summary and guide for study of the theoretical basis of the finite element computer program ADINA Solids & Structures (ADINA in short). The program ADINA is used for displacement and stress analysis.

Since a large number of analysis options is available in this computer program, a user might well be initially overwhelmed with the different analysis choices and the theoretical bases of the computer program. A significant number of publications referred to in the text (books, papers and reports) describe in detail the finite element analysis procedures used in the program. However, this literature is very comprehensive and frequently provides more detail than the user needs to consult for the effective use of ADINA. Furthermore, it is important that a user can identify easily which publication should be studied if more information is desired on a specific analysis option.

The intent with this Theory and Modeling Guide is

- To provide a document that summarizes the methods and assumptions used in the computer program ADINA
- To provide specific references that describe the finite element procedures in more detail.

Hence, this manual has been compiled to provide a bridge between the actual practical use of the ADINA system and the theory documented in various publications. Much reference is made to the book *Finite Element Procedures* (ref. KJB) and to other publications but we endeavored to be specific when referencing these publications so as to help you to find the relevant information.

ref. K. J. Bathe, *Finite Element Procedures*, 2nd ed.,
Cambridge, MA: Klaus-Jürgen Bathe, 2014.

Following this introductory chapter, Chapter 2 describes the elements available in ADINA. The formulations used for these elements have been proven to be reliable and efficient in linear, large displacement, and large strain analyses. Chapter 3 describes

the material models available in ADINA. Chapter 4 describes the different contact formulations and provides modeling tips for contact problems. Loads, boundary conditions and constraints are addressed in Chapter 5. Eigenvalue type analyses such as linearized buckling and frequency analyses are described in Chapter 6. Chapter 7 provides the formulations used for static and implicit dynamic analysis, while Chapter 8 deals with explicit dynamic analysis. Several frequency domain analysis tools are detailed in Chapter 9. Fracture mechanics features are described in Chapter 10. Additional capabilities such as substructures, cyclic symmetry, initial conditions, parallel processing and restarts are discussed in Chapter 11. Chapter 12 describes the heat transfer capabilities in the ADINA program. Finally, some post-processing considerations are provided in Chapter 13.

We intend to update this report as we continue our work on the ADINA system. If you have any suggestions regarding the material discussed in this manual, we would be glad to hear from you.

1.2 Supported computers and operating systems

Table 1.2-1 shows all supported computers/operating systems.

1.3 Units

When using the ADINA system, it is important to enter all physical quantities (lengths, forces, masses, times, etc.) using a consistent set of units. For example, when working with the SI system of units, enter lengths in meters, forces in Newtons, masses in kg, times in seconds. When working with other systems of units, all mass and mass-related units must be consistent with the length, force and time units. For example, in the USCS system (USCS is an abbreviation for the U.S. Customary System), when the length unit is inches, the force unit is pound and the time unit is second, the mass unit is $\text{lb-sec}^2/\text{in}$, not lb.

Rotational degrees of freedom are always expressed in radians. Most angular input parameters are expressed in degrees.

Table 1.3-1 gives some of the more commonly used units needed for ADINA input.

Table 1.2-1: Supported computers/operating systems

Platform	Operating system	Fortran compiler
Linux x86_64	2.6.18 and higher, glibc 2.5 and higher, gcc 4.1.2 and higher	Intel ifort 11.1
Linux 86_64, AVX extensions	2.6.32 and higher, glibc 2.12 and higher, gcc 4.4.6 and higher	Intel ifort 14.0.4
Windows x86_64	Windows XP, Vista, 7, 8, 8.1, 10	Intel Visual Fortran 11
Windows x86_64, AVX extensions	Windows 7, 8, 8.1, 10	Intel Visual Fortran 14

- 1) All program versions are 64-bit, using the x86_64 architecture. The Intel 64 and AMD Opteron implementations of the x86_64 architecture are supported.

Table 1.3-1: Units

	SI	SI (mm)	USCS (inch)	USCS (kip)
length	meter (m)	millimeter (mm)	inch (in)	inch (in)
force	Newton (N)	Newton (N)	pound (lb)	kip (1000 lb)
time	second (s)	second (s)	second (sec)	second (sec)
mass	kilogram (kg) = N-s ² /m	N-s ² /mm	lb-sec ² /in (note 1)	kip-sec ² /in
pressure, stress, Young's modulus, etc.	Pascal (Pa) = N/m ²	N/mm ²	psi = lb/in ²	ksi = kip/in ²
density	kg/m ³	N-s ² /mm ⁴	lb-sec ² /in ⁴ (note 2)	kip-sec ² /in ⁴

1) A body that weighs 1 lb has a mass of $1/386.1 = 0.002590$ lb-sec²/in.

2) A body with weight density 1 lb/in³ has a density of 0.002590 lb-sec²/in⁴.

1.4 ADINA System documentation

At the time of printing of this manual, the following documents are available with the ADINA System:

Installation Notes

Describes the installation of the ADINA System on your computer. Depending on the platform, the appropriate installation notes in pdf format can be printed or downloaded from <http://www.adina.com>

ADINA Handbook

Written as a task-oriented desktop reference, the ADINA handbook helps users to quickly and effectively leverage ADINA's advanced geometric modeling, meshing, and visualization features.

ADINA User Interface Command Reference Manual

Volume I: ADINA Solids & Structures Model Definition

Volume II: ADINA Thermal Model Definition

Volume III: ADINA CFD & FSI Model Definition

Volume IV: ADINA EM Model Definition

Volume V: Display Processing

These documents describe the AUI command language. You use the AUI command language to write batch files for the AUI.

ADINA Primer

Tutorial for the ADINA User Interface, presenting a sequence of worked examples which progressively instruct you how to effectively use the ADINA System.

Theory and Modeling Guide

Volume I: ADINA Solids & Structures

Volume II: ADINA Thermal

Volume III: ADINA CFD & FSI

Volume IV: ADINA EM

Provides a concise summary and guide for the theoretical basis of the analysis programs. The manuals also provide references to other publications which contain further information, but the detail contained in the manuals is usually sufficient for effective understanding and use of the programs.

ADINA Verification Manual

Presents solutions to problems which verify and demonstrate the usage of the ADINA System. Input files for these problems are distributed along with the ADINA System programs.

ADINA-Nastran Interface Manual

Describes the ADINA-AUI Nastran Interface. This guide is available as a pdf file. The Nastran Case Control Commands, Parameters, and Bulk Data Entries that are supported by the AUI are documented.

TRANSOR for I-DEAS Users Guide

Describes the interface between the ADINA System and NX I-deas. This guide is available in html format and is directly accessible from the TRANSOR interface within I-deas. The use of TRANSOR for I-deas to perform pre-/post-processing and ADINA analysis within the I-deas environment, is described.

TRANSOR for Femap Users Guide

Describes the interface between the ADINA System and Femap. This guide is available as a pdf file. The use of TRANSOR for Femap to perform pre-/post-processing and ADINA analysis within the Femap environment, is described.

ADINA System 9.6 Release Notes

Provides a description of the new and modified features of the ADINA System 9.6.

This page intentionally left blank

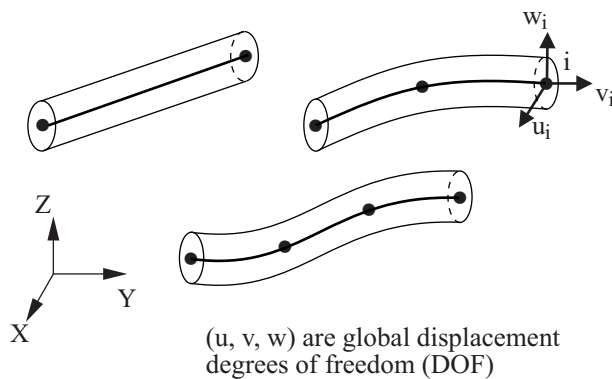
2. Elements

2.1 Truss and cable elements

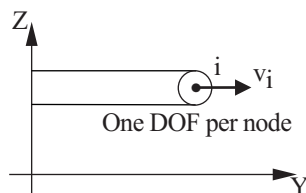
- This chapter outlines the theory behind the different element classes, and also provides details on how to use the elements in modeling. This includes the materials that can be used with each element type, their applicability to large displacement and large strain problems, their numerical integration, etc.

2.1.1 General considerations

- The truss elements can be employed as 2-node, 3-node and 4-node elements, or as a 1-node ring element. Fig. 2.1-1 shows the elements available in ADINA.



(a) 2-, 3- and 4-node elements



(b) 1-node ring element

ref. KJB
Sections
5.3.1,
6.3.3

Figure 2.1-1: Truss elements available in ADINA

- Note that the only force transmitted by the truss element is the longitudinal force as illustrated in Fig. 2.1-2. This force is constant in the 2-node truss and the ring element, but can vary in the 3- and 4-node truss (cable) elements.

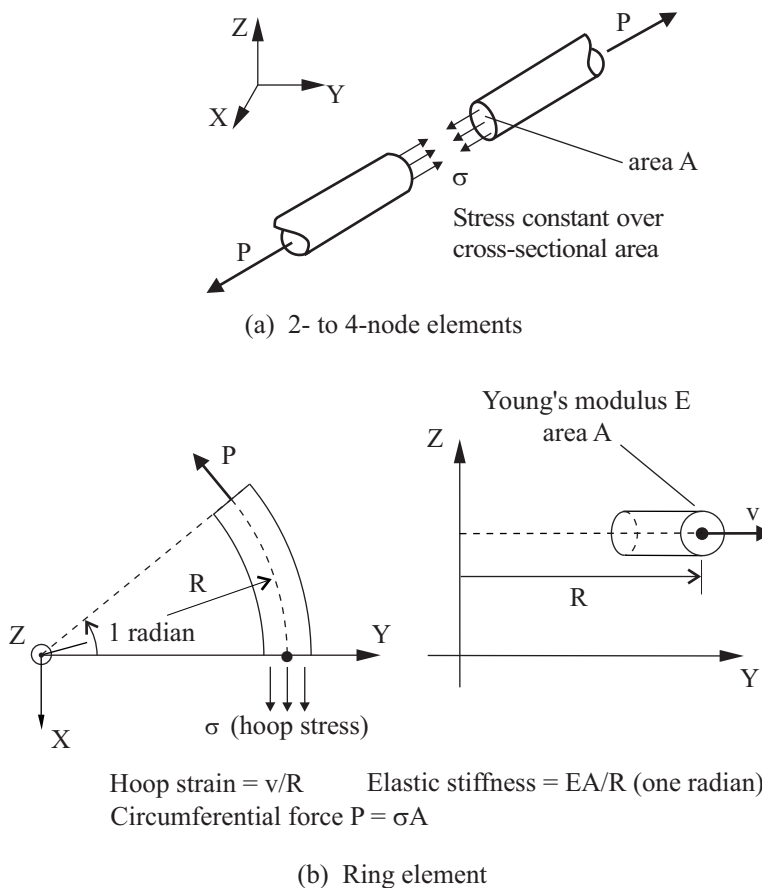


Figure 2.1-2: Stresses and forces in truss elements

- The ring element is formulated for one radian of the structure (as is the axisymmetric 2-D solid element, see Section 2.2.1), and this formulation is illustrated in Fig. 2.1-2.
- The local node numbering and natural coordinate system for the 2-node, 3-node and 4-node truss elements are shown in Fig. 2.1-3.

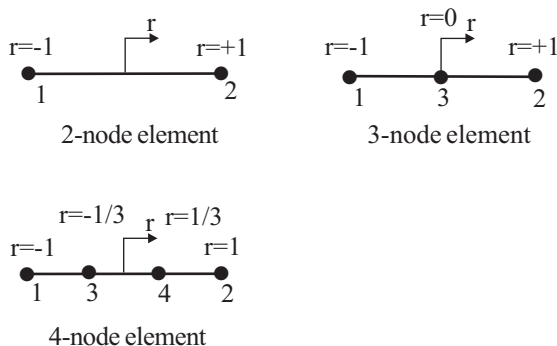


Figure 2.1-3: Local node numbering; natural coordinate system

2.1.2 Material models and formulations

- The truss element can be used with the following material models: **elastic-isotropic, nonlinear-elastic, plastic-bilinear, plastic-multilinear, plastic-cyclic, thermo-isotropic, thermo-plastic, creep, plastic-creep, multilinear-plastic-creep, creep-variable, plastic-creep-variable, multilinear-plastic-creep-variable, viscoelastic, shape-memory alloy, user-coded.**
- The truss element can be used with the **small** or **large displacement** formulations. In the small displacement formulation, the displacements and strains are assumed infinitesimally small. In the large displacement formulation, the displacements and rotations can be very large. In all cases, the cross-sectional area of the element is assumed to remain unchanged, and the strain is equal to the longitudinal displacement divided by the original length.

All of the material models in the above list can be used with either formulation. The use of a linear material with the small displacement formulation corresponds to a linear formulation, and the use of a nonlinear material with the small displacement formulation corresponds to a materially-nonlinear-only formulation.

2.1.3 Numerical integration

- The integration point labeling of the truss (cable) elements is given in Fig. 2.1-4.

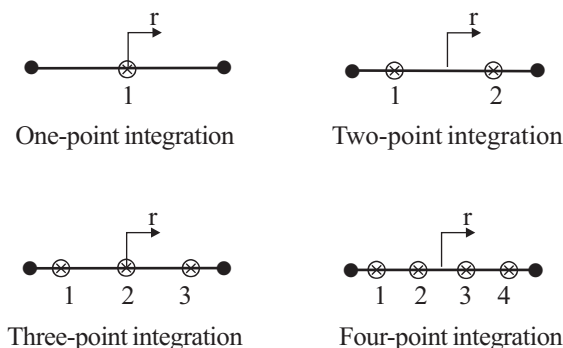


Figure 2.1-4: Integration point locations

- Note that when the material is temperature-independent, the 2-node truss and ring elements require only 1-point Gauss numerical integration for an exact evaluation of the stiffness matrix, because the force is constant in the element. However, 2-point Gauss numerical integration may be appropriate when a temperature-dependent material is used because, due to a varying temperature, the material properties can vary along the length of an element.

2.1.4 Mass matrices

- The consistent mass matrix is calculated using Eq. (4.25) in ref. KJB, p. 165, which accurately computes the consistent mass distribution.

The lumped mass matrix is formed by dividing the element's mass among its nodes. The mass assigned to each node is $M \cdot \left(\frac{\ell_i}{L} \right)$,

where M = total mass, L = total element length, ℓ_i = fraction of the total element length associated with element node i (i.e., for the 2-node truss element, $\ell_1 = \frac{L}{2}$ and $\ell_2 = \frac{L}{2}$, and for the 4-node truss $\ell_1 = \ell_2 = \frac{L}{6}$ and $\ell_3 = \ell_4 = \frac{L}{3}$). The element has no rotational mass.

2.1.5 Element output

- Each element outputs, at its integration points, the following information to the porthole file, based on the material model. This information is accessible in the AUI using the given variable names.

Elastic-isotropic, nonlinear-elastic: FORCE-R, STRESS-RR, STRAIN-RR

Elastic-isotropic with thermal effects: FORCE-R, STRESS-RR, STRAIN-RR, THERMAL_STRAIN, ELEMENT_TEMPERATURE

Thermo-isotropic: FORCE-R, STRESS-RR, STRAIN-RR, ELEMENT_TEMPERATURE

Plastic-bilinear, plastic-multilinear: PLASTIC_FLAG, FORCE-R, STRESS-RR, STRAIN-RR, PLASTIC_STRAIN-RR

SMA: EFFECTIVE_STRESS, STRESS-RR, ACCUM_EFF_TRANSF_STRAIN, AUSTENITE_FRACTION, DETWINNED_MARTENSITE_FRACTION, STRAIN-RR, THERMAL_STRAIN-RR, TRANSFORMATION_STRAIN-RR, TWINNED_MARTENSITE_FRACTION, SMA_FLAG

Thermo-plastic, creep, plastic-creep, multilinear-plastic-creep: PLASTIC_FLAG, NUMBER_OF_SUBINCREMENTS, FORCE-R, STRESS-RR, STRAIN-RR, ACCUMULATED_EFFECTIVE_PLASTIC_STRAIN, PLASTIC_STRAIN-RR, CREEP_STRAIN-RR, THERMAL_STRAIN-RR, ELEMENT_TEMPERATURE, FE_EFFECTIVE_STRESS, YIELD_STRESS, EFFECTIVE_CREEP_STRAIN

Viscoelastic: PLASTIC_FLAG, STRESS-RR, STRAIN-RR, THERMAL_STRAIN-RR, ELEMENT_TEMPERATURE

User-coded: STRESS-RR, STRAIN-RR, USER_VARIABLE_I, INT_USER_VARIABLE_I

See Section 13.1.1 for the definitions of those variables that are not self-explanatory.

2.1.6 Recommendations on use of elements

- Of the 2-, 3- and 4-node truss elements, the 2-node element is usually most effective and can be used in modeling truss structures, cable structures, linear springs (the spring stiffness k is equal to EA/L where L is the length of the element) and nonlinear elastic gap elements (see Section 3.3).
- The 3- and 4-node elements are employed to model cables and steel reinforcement in reinforced concrete structures that are modeled with higher-order continuum or shell elements. In this case, the 3- and 4-node truss elements are compatible with the continuum or shell elements.
- The internal nodes for the 3-node and 4-node truss elements are usually best placed at the mid- and third-points, respectively, unless a specific predictive capability of the elements is required (see ref. KJB, Examples 5.2 and 5.17, pp. 348 and 370).
- The 1-node ring element is formulated with a single degree of freedom, corresponding to a displacement in the Y direction. Therefore the 1-node ring element has no stiffness or mass in the Z direction. In addition, if a skew system is assigned to the ring element node, the skew system is ignored by the ring element.

2.1.7 Rebar elements

- The AUI can generate truss elements that are connected to the 2D or 3D solid elements in which the truss elements lie. There are several separate cases, corresponding to the type of the truss element group and the type of the solid element group in which the truss elements lie.

Axisymmetric truss elements, axisymmetric solid elements: The AUI can connect axisymmetric truss elements that exist in the model prior to data file generation to the axisymmetric 2D elements in which the truss elements lie. The AUI does this connection

during data file generation as follows. For each axisymmetric truss element that lies in an axisymmetric 2D element, constraint equations are defined between the axisymmetric truss element node and the three closest corner nodes of the 2D element.

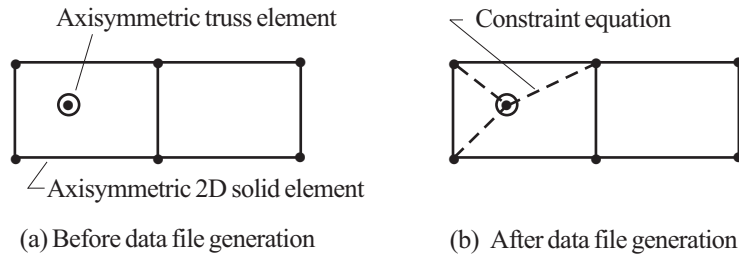


Figure 2.1-5: Rebar in axisymmetric truss, 2D solid elements

3D truss elements, planar 2D solid elements: The AUI can generate 3D truss elements and then connect the truss elements to the planar 2D elements in which the truss elements lie. The AUI does this during data file generation as follows. For each rebar line, the AUI finds the intersections of the rebar line and the sides of the 2D elements. The AUI then generates nodes at these intersections and generates truss elements that connect the successive nodes. The AUI also defines constraint equations between the generated nodes and the corner nodes of the 2D elements.

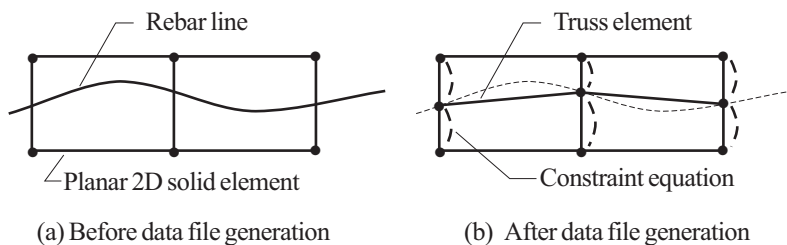


Figure 2.1-6: Rebar in 3D truss, planar 2D solid elements

3D truss elements, 3D solid elements: The AUI can generate 3D truss elements and then connect the truss elements to the 3D elements in which the truss elements lie. The AUI does this during data file generation as follows. For each rebar line, the AUI finds the intersections of the rebar line and the faces of the 3D elements.

The AUI then generates nodes at these intersections and generates truss elements that connect the successive nodes. The AUI also defines constraint equations between the generated nodes and three closest corner nodes of the 3D element faces.

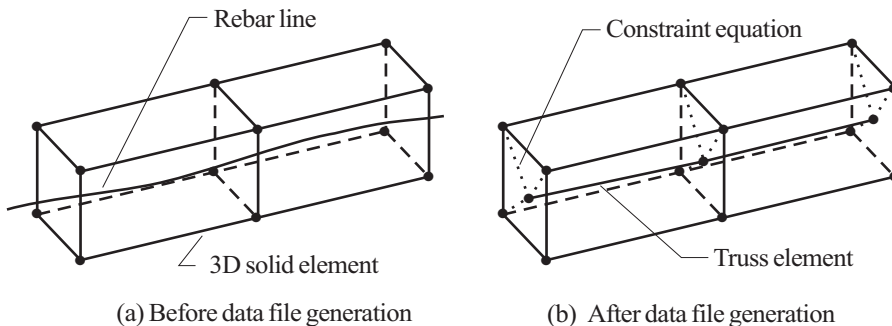


Figure 2.1-7: Rebar in 3D truss, 3D solid elements

- The rebar option is intended for use with lower-order solid elements (that is, the solid elements should not have mid-side nodes).
- For 3D truss element generation, if the rebar lines are curved, you should specify enough subdivisions on the rebar lines so that the distance between subdivisions is less than the solid element size. The more subdivisions that are on the rebar lines, the more accurately the AUI will compute the intersections of the rebar lines and the solid element sides or faces.
- To request the connection:

AUI: Click the Define Element Group icon to open the Define Element Group window. In this window, check that the Element Type has been set to “Truss”. Click on the Advanced tab. Select the “Use as Rebars” option from the Element Option drop-down, and then fill in the Rebar-Line Label box below the Element Option drop-down box.

Command Line: Set `OPTION=REBAR` in the `EGROUP TRUSS` command. For 3D truss elements, use the `REBAR-LINE` command and `RB-LINE` parameter of the `EGROUP TRUSS` command to define the rebar lines.

2.2 Two-dimensional solid elements

2.2.1 General considerations

- The following kinematic assumptions are available for two-dimensional elements in ADINA: **plane stress**, **plane strain**, **axisymmetric**, **generalized plane strain** and **3-D plane stress** (membrane). Figures 2.2-1 and 2.2-2 show some typical 2-D elements and the assumptions used in the formulations.

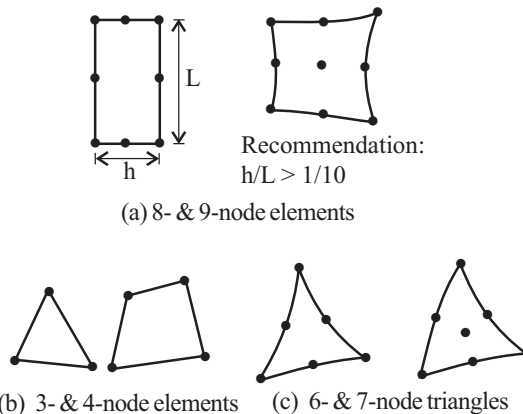


Figure 2.2-1: 2-D solid elements

- The plane stress, plane strain, generalized plane strain, and axisymmetric elements can be defined in the X-Y, Y-Z or X-Z planes. The axisymmetric element can have x or y or z as the rotating axis and does not need to be located in the +X, +Y or +Z half planes.
- The 3-D plane stress element can lie in general three-dimensional space. The element can be initially flat or can be curved. This element is sometimes called a membrane element.
- The axisymmetric element represents one radian of the structure. The stiffness, mass and loads are defined accordingly. Hence, when this element is combined with other elements, or when concentrated loads are defined, these must also refer to one radian, see ref. KJB, Examples 5.9 and 5.10, p. 356.

ref. KJB
Sections 5.3.1
and 5.3.2

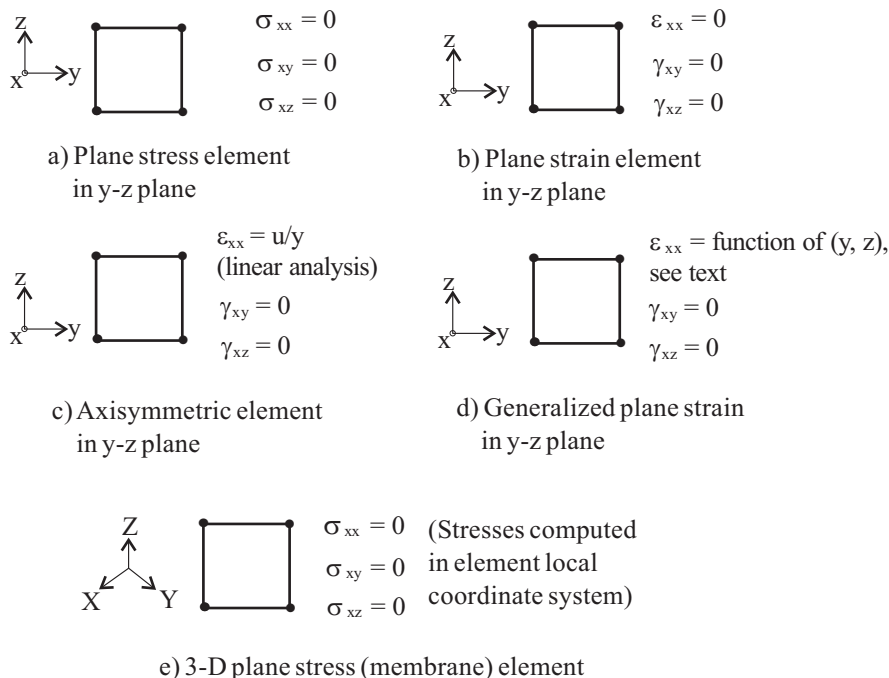


Figure 2.2-2: Basic assumptions in 2-D analysis

- The plane strain element represents a unit thickness of the structure. The stiffness, mass and loads are defined accordingly.
- The plane stress element is either a constant thickness element or a varying thickness element. In a varying thickness element, the thickness at each element local node can be different.
- The generalized plane strain element is intended to be used to model beam-like structures which have a constant cross-section and are submitted to a loading independent of the axial coordinate of the beam. In such cases, the deformations of the structure are independent of the axial coordinate of the structure. This situation is shown in Fig. 2.2-7(d).

The solution is valid only for the section of the structure that are far enough from the ends; otherwise local 3-D effects usually render the generalized plane strain assumptions invalid.

- The elements usually used in ADINA are isoparametric

displacement-based finite elements, and their formulation is described in detail in ref. KJB, Section 5.3.

- The basic finite element assumptions are, see Fig. 2.2-3, for the coordinates

$$y = \sum_{i=1}^q h_i y_i ; \quad z = \sum_{i=1}^q h_i z_i$$

for the displacements in plane stress, plane strain, and axisymmetric elements

$$v = \sum_{i=1}^q h_i v_i ; \quad w = \sum_{i=1}^q h_i w_i$$

for the displacements in generalized plane strain elements

$$u = xU^p - x(y - y_p)\Phi_z^p + x(z - z_p)\Phi_y^p$$

$$v = \sum_{i=1}^q h_i v_i + \frac{1}{2}x^2\Phi_z^p \quad w = \sum_{i=1}^q h_i w_i - \frac{1}{2}x^2\Phi_y^p$$

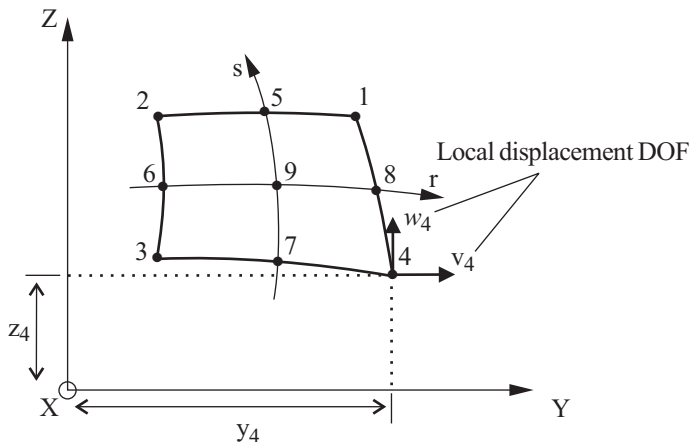
where

- $h_i(r,s)$ = interpolation function corresponding to node i
- (r,s) = isoparametric coordinates
- q = number of element nodes, $3 \leq q \leq 9$, excluding auxiliary node in generalized plane strain
- y_i, z_i = nodal point coordinates
- v_i, w_i = nodal point displacements
- U^p, Φ_y^p, Φ_z^p = degrees of freedom of the auxiliary node
- x_p, y_p, z_p = coordinates of the auxiliary node

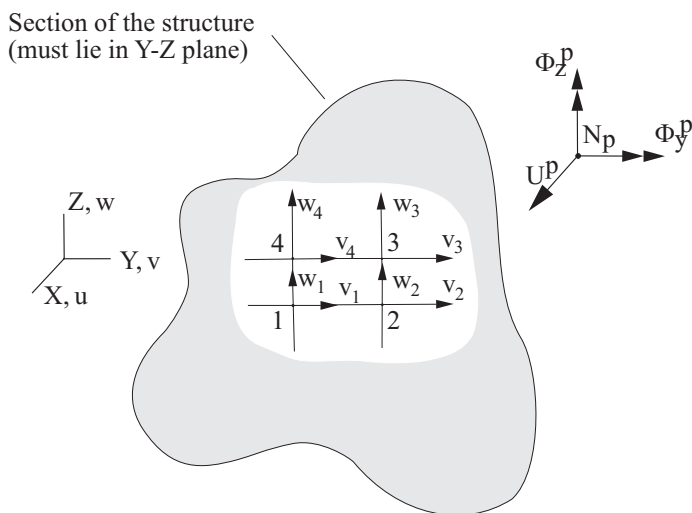
- For generalized plane strain the elements are assumed to have a constant curvature in the X-direction, and to lie between two planes that can move relative to each other. One plane must be the $X = 0$ plane where the elements must be located. The location of the other plane is specified via the auxiliary point which must lie on that plane.

An axial force and bending moments can be applied to the auxiliary point, and these forces/moments can be used to model the

corresponding forces/moments acting onto the beam structure.



a) Plane strain, plane stress, and axisymmetric elements



The coordinates of the auxiliary node N_p are (x_p, y_p, z_p) .

The element has unit length in the X direction.

b) Generalized plane strain element

Figure 2.2-3: Conventions used for the nodal coordinates and displacements of the 2-D solid element

The displacements in the element can be interpreted as the sum

of the 2-D in-plane displacements (due to in-plane forces and boundary conditions) plus additional displacements which come from the out-of-plane rigid body motions of the cross-section (due to the axial force and bending moments applied to the structure).

From the displacement interpolations, the out-of-plane normal strain is seen to be

$$\varepsilon_{xx} = U^p - (y - y_p)\Phi_z^p + (z - z_p)\Phi_y^p$$

- In addition to the displacement-based elements, special mixed-interpolated elements are also available, in which the displacements and pressure are interpolated separately. These elements are effective and should be preferred in the analysis of incompressible media and inelastic materials (specifically for materials in which Poisson's ratio is close to 0.5, for rubber-like materials and for elasto-plastic materials). The mixed formulation is only available for the plane strain, axisymmetric and generalized plane strain elements. It is not available (and not needed) for plane stress and 3-D plane stress (membrane) elements.
- The mixed formulation is the default setting for the materials in which either the entire response is incompressible (for example, the rubber material models), or in which the inelastic response is incompressible (for example, the plastic and creep material models).
- Table 2.2-1 shows the default number of pressure degrees of freedom for each 2-D element type. For more details on the mixed interpolation of pressure and displacement degrees of freedom for 3-D solids, see Section 4.4.3, p. 276, and Tables 4.6 and 4.7, pp.292-295 in ref. KJB.

Table 2.2-1: Mixed formulation settings for 2-D solid elements

2-D solid element	Number of default pressure DOFs
3-node triangle	1*
4-node quadrilateral (4/1)	1
6-node triangle (6/3)	3
7-node triangle (7/3)	3
8-node quadrilateral (8/3)	3
9-node quadrilateral (9/3)	3

* To use this element, the user must use the displacement-based element with 1-point integration.

- The default number of pressure degrees of freedom shown in Table 2.2-1 is optimal for most applications. However, it can be modified if needed. In some problems where rubber is subjected to severe compression, numerical instability can be avoided by using 4 pressure degrees of freedom for the 8 and 9-node quadrilaterals.
- The following choices for the number of pressure degrees of freedom are valid; 1 for a constant pressure, 3 for additional linear pressure terms, 4 for an additional bilinear pressure term (involving r s), and 6 for additional quadratic pressure terms (involving r^2 , s^2).
- In general, using too many pressure degrees of freedom leads to volumetric locking, while using too few pressure degrees of freedom leads to spurious zero-energy modes. See Sections 4.4.2 and 4.5 in ref. KJB for more details.
 - ref. T. Sussman and K.J. Bathe, "A Finite Element Formulation for Nonlinear Incompressible Elastic and Inelastic Analysis," *J. Computers and Structures*, Vol. 26, No. 1/2, pp. 357-409, 1987.
- In addition to the displacement-based and mixed-interpolated elements, ADINA also includes the possibility of including incompatible modes in the formulation of the 4-node element.

Within this element, additional displacement degrees of freedom are introduced. These additional displacement degrees of freedom are not associated with nodes; therefore the condition of displacement compatibility between adjacent elements is not satisfied in general. However the additional displacement degrees of freedom increase the flexibility of the element, especially in bending situations.

Figure 2.2-4 shows a situation in which the incompatible modes elements give an improved solution.

For theoretical considerations, see reference KJB, Section 4.4.1. Note that these elements are formulated to pass the patch test. Also note that element distortions deteriorate the element performance.

The incompatible modes feature cannot be used in conjunction with the mixed-interpolation formulation.

The incompatible modes feature is only available for the 4-node element; in particular note that the incompatible modes feature is not available for the 3-node triangular element.

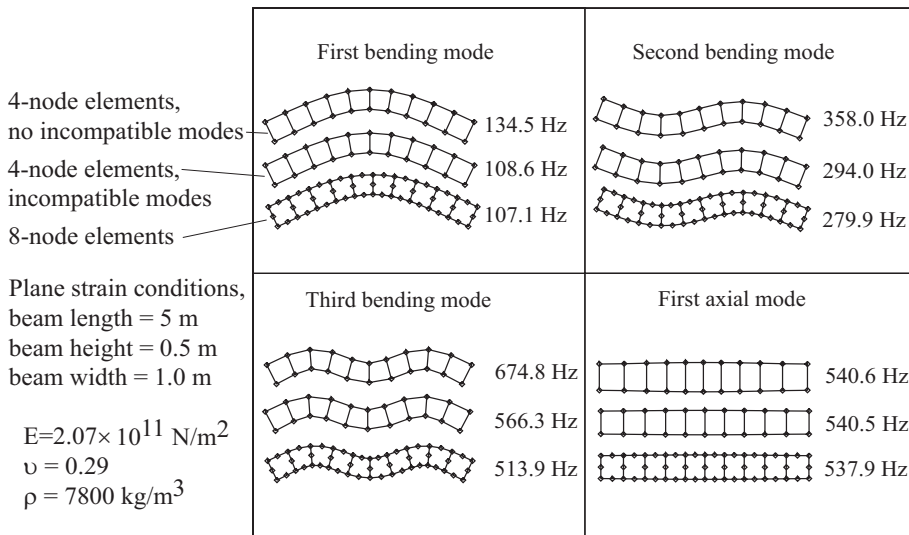


Figure 2.2-4: Frequency analysis of a free-free beam with three types of finite elements

- The elements can be used with 3 to 9 nodes. The interpolation functions are defined in ref. KJB, Fig. 5.4, p. 344.

ref. KJB
Section 5.3.2

- Triangular elements are formed in ADINA by assigning the

same global node number to all the nodes located on one side. It is recommended that this "collapsed" side be the one with local end nodes 1 and 4.

The triangular elements can be of different types:

- ▶ A collapsed 6-node triangular element with the strain singularity $\frac{1}{\sqrt{r}}$ (see Chapter 10)
- ▶ A collapsed 8-node element with the strain singularity $\frac{1}{r}$ (see Chapter 10)
- ▶ A collapsed 6-node spatially isotropic triangular element
- ▶ A collapsed 7-node spatially isotropic triangular element

These element types are described on pp. 369-370 of ref. KJB. The strain singularities are obtained by collapsing one side of an 8-node rectangular element and not correcting the interpolation functions.

The 6-node spatially isotropic triangle is obtained by correcting the interpolation functions of the collapsed 8-node element. It then uses the same interpolation functions for each of the 3 corner nodes and for each of the midside nodes. Note that when the corrections to the interpolation functions are employed, only the element side with end nodes 1 and 4 can be collapsed.

The 3-node triangular element is obtained by collapsing one side of the 4-node element. This element exhibits the constant strain conditions (except that the hoop strain in axisymmetric analysis varies over the element) but is usually not effective.

- Linear and nonlinear fracture mechanics analysis of stationary or propagating cracks can be performed with two-dimensional elements (see Chapter 10).

2.2.2 Material models and formulations

- The 2-D elements can be used with the following material models: **elastic-isotropic**, **elastic-orthotropic** (with or without wrinkling), **nonlinear-elastic**, **plastic-bilinear**, **plastic-**

multilinear, Drucker-Prager, Mroz-bilinear, plastic-orthotropic, thermo-isotropic, thermo-orthotropic, thermo-plastic, piezoelectric, creep, plastic-creep, multilinear-plastic-creep, concrete, curve-description, Ogden, Mooney-Rivlin, Arruda-Boyce, hyper-foam, Sussman-Bathe, creep-variable, plastic-creep-variable, multilinear-plastic-creep-variable, Gurson-plastic, Cam-clay, Mohr-Coulomb, viscoelastic, creep irradiation, gasket, shape-memory alloy, plastic-cyclic, user-coded.

- In plane strain, generalized plane strain and axisymmetric analysis, the mixed interpolation formulation should be used (and is the default) whenever using the plastic-bilinear, plastic-multilinear, Mroz-bilinear, plastic-orthotropic, thermo-plastic, creep, plastic-creep, multilinear-plastic-creep, Ogden, Mooney-Rivlin, Arruda-Boyce, Sussman-Bathe, creep-variable, plastic-creep-variable, multilinear-plastic-creep-variable, plastic-cyclic or viscoelastic material models.

In addition the mixed interpolation formulation should be used (but is not the default) when using the elastic-isotropic material and the Poisson's ratio is close to 0.5.

- The two-dimensional elements can be used with a **small displacement/small strain, large displacement/small strain** or a **large displacement/large strain** formulation.

The small displacement/small strain and large displacement/small strain formulations can be used with any material model, except for the Ogden, Mooney-Rivlin, Arruda-Boyce, hyper-foam and Sussman-Bathe material models. The use of a linear material with the small displacement/small strain formulation corresponds to a linear formulation, and the use of a nonlinear material with the small displacement/small strain formulation corresponds to a materially-nonlinear-only formulation. The program uses the TL (total Lagrangian) formulation when you choose a large displacement/small strain formulation.

The large displacement/large strain formulations can be used with the plastic-bilinear, plastic-multilinear, Mroz-bilinear, plastic-orthotropic, thermo-plastic, creep, plastic-creep, multilinear-plastic-creep, creep-variable, plastic-creep-variable, multilinear-plastic-creep-variable, viscoelastic, plastic-cyclic and user-coded material

models. The program uses either a ULH or ULJ formulation when you choose a large displacement/large strain formulation with these material models.

A large displacement/large strain formulation is used with the Ogden, Mooney-Rivlin, Arruda-Boyce, hyper-foam and Sussman-Bathe material models. The program uses a TL (total Lagrangian) formulation in this case.

ref. KJB
Sections 6.2 and
6.3.4

- The basic continuum mechanics formulations are described in ref. KJB, pp. 497-537, and the finite element discretization is given in ref. KJB pp. 538-542, 549-555.

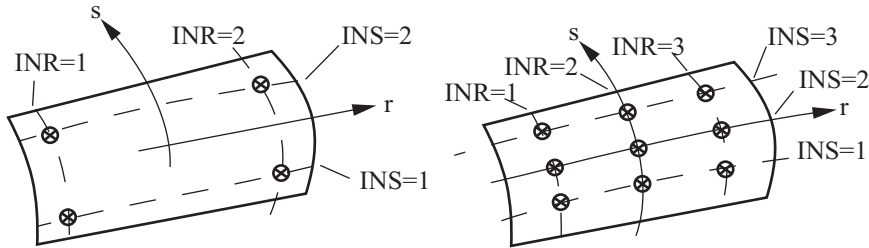
ref. KJB
Section 6.8.1

- Note that all these formulations can be used together in a single finite element mesh. If the elements are initially compatible, then they will remain compatible throughout the complete analysis.

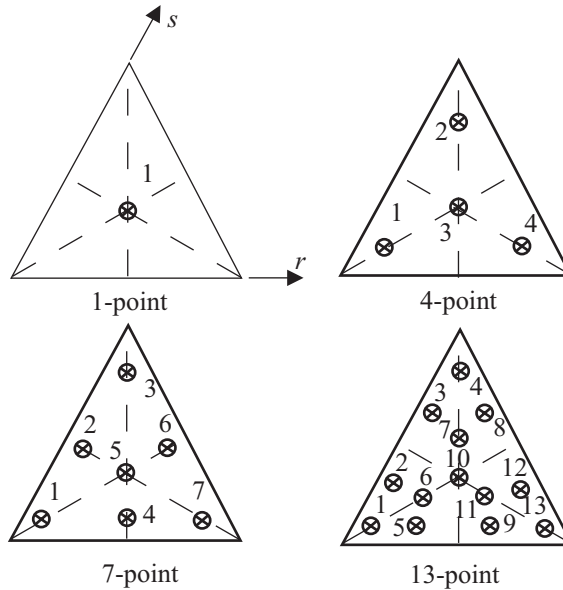
2.2.3 Numerical integration

ref. KJB
Sections 5.5.3,
5.5.4 and 5.5.5

- For the calculation of all element matrices and vectors, numerical Gauss integration is used. You can use from 2×2 to 6×6 Gauss integration. The numbering convention and the location of the integration points are given in Fig. 2.2-5 for 2×2 and 3×3 Gauss integration. The convention used for higher orders is analogous. The default Gauss integration orders for rectangular elements are 2×2 for 4-node elements and 3×3 otherwise.



a) Rectangular elements



b) Triangular elements

Figure 2.2-5: Integration point positions for 2-D solid elements

- For the 8-node rectangular element, the use of 2×2 Gauss integration corresponds to a slight under-integration and one spurious zero energy mode is present. In practice, this particular kinematic mode usually does not present any problem in linear analysis (except in the analysis of one-element cases, see ref. KJB, Fig. 5.40, p. 472) and thus 2×2 Gauss integration can be employed with caution for the 8-node elements.

- The 3-node, 6-node and 7-node triangular elements are spatially isotropic with respect to integration point locations and interpolation functions (see Figure 2.2-5).

- The integration order used for triangular elements depends upon the integration order used for rectangular elements as follows.

When rectangular elements use 2×2 Gauss integration, triangular elements use 4-point Gauss integration; when rectangular elements use 3×3 Gauss integration, triangular elements use 7-point Gauss integration; when rectangular elements use 4×4 Gauss integration or higher, triangular elements use 13-point Gauss integration.

However, for 3-node triangular elements, when the axisymmetric subtype is not used, then 1-point Gauss integration is always used.

- Note that in geometrically nonlinear analysis, the spatial positions of the Gauss integration points change continuously as the element undergoes deformations, but throughout the response the same material particles are at the integration points.

ref. KJB
Section 6.8.4

- The order of numerical integration in the large displacement elastic analysis is usually best chosen to be equal to the order used in linear elastic analysis. Hence, the program default values should usually be employed. However, in inelastic analysis, a higher order integration should be used when the elements are used to model thin structures (see Fig. 6.25, p. 638, ref. KJB), or large deformations (large strains) are anticipated.

2.2.4 Mass matrices

- The consistent mass matrix is always calculated using either 3×3 Gauss integration for rectangular elements or 7-point Gauss integration for triangular elements.
- The lumped mass matrix of an element is formed by dividing the element's mass M equally among its n nodes. Hence, the mass assigned to each node is M/n . No special distributory concepts are employed to distinguish between corner and midside nodes, or to account for element distortion.
- Note that n is the number of distinct non-repeated nodes in the

element. Hence, when a quad element side is collapsed to a single node, the total mass of the element is equally distributed among the final nodes of the triangular configuration.

2.2.5 Element output

You can request that ADINA either print or save stresses or forces.

Stresses: Each element outputs, at its integration points, the following information to the porthole file, based on the material model. This information is accessible in the AUI using the given variable names.

Notice that for the orthotropic and piezoelectric material models, you can request that the stresses and strains be saved in the material coordinate system.

For the 3-D plane stress (membrane) elements, results with indices YY, ZZ, etc are saved in the element local coordinate system (see Figure 2.2-6).

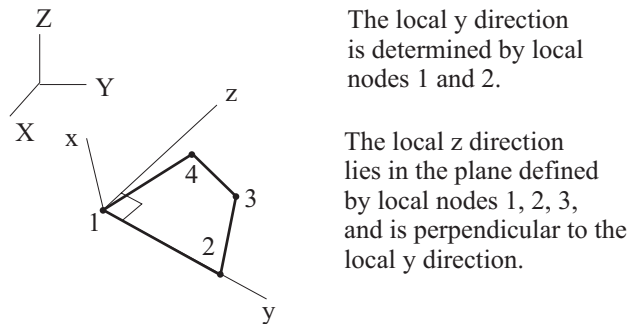


Figure 2.2-6: Local coordinate system for 3-D plane stress (membrane) element

Elastic-isotropic, elastic-orthotropic (no wrinkling, results saved in global system): STRESS (XYZ) , STRAIN (XYZ)

Elastic-isotropic with thermal effects: STRESS (XYZ) ,
STRAIN (XYZ) , THERMAL_STRAIN,
ELEMENT_TEMPERATURE

Elastic-orthotropic (no wrinkling, results saved in material system):
STRESS (ABC) , STRAIN (ABC)

Thermo-isotropic, thermo-orthotropic (results saved in global system): STRESS (XYZ) , STRAIN (XYZ) ,
THERMAL_STRAIN (XYZ) , ELEMENT_TEMPERATURE

Piezoelectric (results saved in global system): STRESS (XYZ) ,
STRAIN (XYZ) , ELECTRIC_FIELD-Y,
ELECTRIC_FIELD-Z,
ELECTRIC_DISPLACEMENT_FIELD-Y,
ELECTRIC_DISPLACEMENT_FIELD-Z.

Piezoelectric (results saved in material system) :
STRESS (ABC) , STRAIN (ABC) , ELECTRIC_FIELD-A,
ELECTRIC_FIELD-B,
ELECTRIC_DISPLACEMENT_FIELD-A,
ELECTRIC_DISPLACEMENT_FIELD-B.

Curve description: CRACK_FLAG, STRESS (XYZ) ,
STRAIN (XYZ) , FE_SIGMA-P1, FE_SIGMA-P2,
FE_SIGMA-P1_ANGLE, GRAVITY_IN-SITU_PRESSURE,
VOLUMETRIC_STRAIN

Concrete: CRACK_FLAG, STRESS (XYZ) , STRAIN (XYZ) ,
FE_SIGMA-P1, FE_SIGMA-P2, FE_SIGMA-P1_ANGLE,
FAILURE_STRESS, ELEMENT_TEMPERATURE,
THERMAL_STRAIN (XYZ)

Small strains: Plastic-bilinear, plastic-multilinear, Mroz-bilinear:
PLASTIC_FLAG, STRESS (XYZ) , STRAIN (XYZ) ,
PLASTIC_STRAIN (XYZ) , FE_EFFECTIVE_STRESS,
YIELD_STRESS,
ACCUMULATED_EFFECTIVE_PLASTIC_STRAIN,
THERMAL_STRAIN (XYZ) , ELEMENT_TEMPERATURE

SMA: EFFECTIVE_STRESS, STRESS-XX, STRESS-YY,
STRESS-YZ, STRESS-ZZ,
ACCUM EFF TRANSF STRAIN, AUSTENITE_FRACTION,
DETWINNED_MARTENSITE_FRACTION, STRAIN-XX,
STRAIN-YY, STRAIN-YZ, STRAIN-ZZ,
THERMAL_STRAIN-XX, THERMAL_STRAIN-YY,
THERMAL_STRAIN-ZZ, TRANSFORMATION_STRAIN-XX,
TRANSFORMATION_STRAIN-YY,

TRANSFORMATION_STRAIN-YZ,
TRANSFORMATION_STRAIN-ZZ,
TWINNED_MARTENSITE_FRACTION, SMA_FLAG

Large strains: Plastic-bilinear, plastic-multilinear, Mroz-bilinear:

PLASTIC_FLAG, STRESS(XYZ),
DEFORMATION_GRADIENT(XYZ),
FE_EFFECTIVE_STRESS, YIELD_STRESS,
ACCUMULATED_EFFECTIVE_PLASTIC_STRAIN,
THERMAL_STRAIN(XYZ), ELEMENT_TEMPERATURE

Small strains: Anand Model

STRESS(XYZ), STRAIN(XYZ),
VISCOPLASTIC_STRAIN(XYZ),
FE_EFFECTIVE_STRESS,
ACCUMULATED_EFFECTIVE_VISCOPLASTIC_STRAIN,
EFFECTIVE_VISCOPLASTIC_STRAIN_RATE,
THERMAL_STRAIN(XYZ), ELEMENT_TEMPERATURE

Large strains: Anand Model

STRESS(XYZ), DEFORMATION_GRADIENT(XYZ),
FE_EFFECTIVE_STRESS,
ACCUMULATED_EFFECTIVE_VISCOPLASTIC_STRAIN,
EFFECTIVE_VISCOPLASTIC_STRAIN_RATE,
THERMAL_STRAIN(XYZ), ELEMENT_TEMPERATURE

Plastic-orthotropic (results saved in global system):

PLASTIC_FLAG, STRESS(XYZ), STRAIN(XYZ),
PLASTIC_STRAIN(XYZ), HILL_EFFECTIVE_STRESS,
YIELD_STRESS,
ACCUMULATED_EFFECTIVE_PLASTIC_STRAIN,
THERMAL_STRAIN(XYZ), ELEMENT_TEMPERATURE

Plastic-orthotropic (results saved in material system):

PLASTIC_FLAG, STRESS(ABC), STRAIN(ABC),
PLASTIC_STRAIN(ABC), HILL_EFFECTIVE_STRESS,
YIELD_STRESS,
ACCUMULATED_EFFECTIVE_PLASTIC_STRAIN,
THERMAL_STRAIN(ABC), ELEMENT_TEMPERATURE

Small strains: Drucker-Prager: PLASTIC_FLAG,

PLASTIC_FLAG-2, STRESS(XYZ), STRAIN(XYZ),
PLASTIC_STRAIN(XYZ), CAP_LOCATION,
YIELD_FUNCTION

Large strains: Drucker-Prager: PLASTIC_FLAG,
 PLASTIC_FLAG-2, STRESS(XYZ),
 DEFORMATION_GRADIENT(XYZ), CAP_LOCATION,
 YIELD_FUNCTION

Small strains: Thermo-plastic, creep, plastic-creep, multilinear-
 plastic-creep, creep-variable, plastic-creep-variable, multilinear-
 plastic-creep-variable: PLASTIC_FLAG,
 NUMBER_OF_SUBINCREMENTS, STRESS(XYZ),
 STRAIN(XYZ), PLASTIC_STRAIN(XYZ),
 CREEP_STRAIN(XYZ), THERMAL_STRAIN(XYZ),
 ELEMENT_TEMPERATURE,
 ACCUMULATED_EFFECTIVE_PLASTIC_STRAIN,
 FE_EFFECTIVE_STRESS, YIELD_STRESS,
 EFFECTIVE_CREEP_STRAIN

Large strains: Thermo-plastic, creep, plastic-creep, multilinear-
 plastic-creep, creep-variable, plastic-creep-variable, multilinear-
 plastic-creep-variable: PLASTIC_FLAG,
 NUMBER_OF_SUBINCREMENTS, STRESS(XYZ),
 DEFORMATION_GRADIENT(XYZ),
 FE_EFFECTIVE_STRESS, YIELD_STRESS,
 ACCUMULATED_EFFECTIVE_PLASTIC_STRAIN,
 THERMAL_STRAIN(XYZ), ELEMENT_TEMPERATURE,
 EFFECTIVE_CREEP_STRAIN

Mooney-Rivlin, Ogden, Arruda-Boyce, hyper-foam (strains
 saved): STRESS(XYZ), STRAIN(XYZ)

Mooney-Rivlin, Ogden, Arruda-Boyce, hyper-foam (deformation
 gradients saved): STRESS(XYZ),
 DEFORMATION_GRADIENT(XYZ).

Small strains: User-supplied: STRESS(XYZ), STRAIN(XYZ),
 USER_VARIABLE_I

Large strains: User-supplied: STRESS(XYZ),
 DEFORMATION_GRADIENT(XYZ), USER_VARIABLE_I

Elastic-orthotropic (wrinkling, results saved in global system):
 WRINKLE_FLAG, STRESS(XYZ), STRAIN(XYZ)

Elastic-orthotropic (wrinkling, results saved in material system):
WRINKLE_FLAG, STRESS(ABC), STRAIN(ABC)

Thermo-orthotropic (results saved in material system):
STRESS(ABC), STRAIN(ABC),
THERMAL_STRAIN(ABC), ELEMENT_TEMPERATURE

Small strains: Gurson plastic: PLASTIC_FLAG,
STRESS(XYZ), STRAIN(XYZ),
PLASTIC_STRAIN(XYZ), FE_EFFECTIVE_STRESS,
YIELD_STRESS,
ACCUMULATED_EFFECTIVE_PLASTIC_STRAIN,
THERMAL_STRAIN(XYZ), ELEMENT_TEMPERATURE,
VOID_VOLUME_FRACTION

Large strains: Gurson plastic: PLASTIC_FLAG,
STRESS(XYZ), DEFORMATION_GRADIENT(XYZ),
FE_EFFECTIVE_STRESS, YIELD_STRESS,
ACCUMULATED_EFFECTIVE_PLASTIC_STRAIN,
THERMAL_STRAIN(XYZ), ELEMENT_TEMPERATURE,
VOID_VOLUME_FRACTION

Small strains, Cam-clay: PLASTIC_FLAG, STRESS(XYZ),
STRAIN(XYZ), PLASTIC_STRAIN(XYZ),
YIELD_SURFACE_DIAMETER_P, YIELD_FUNCTION,
MEAN_STRESS, DISTORTIONAL_STRESS,
VOLUMETRIC_STRAIN, VOID_RATIO,
EFFECTIVE_STRESS_RATIO, SPECIFIC_VOLUME

Large strains, Cam-clay: PLASTIC_FLAG, STRESS(XYZ),
DEFORMATION_GRADIENT(XYZ),
YIELD_SURFACE_DIAMETER_P, YIELD_FUNCTION,
MEAN_STRESS, DISTORTIONAL_STRESS,
VOLUMETRIC_STRAIN, VOID_RATIO,
EFFECTIVE_STRESS_RATIO, SPECIFIC_VOLUME

Mohr-Coulomb: PLASTIC_FLAG, STRESS(XYZ),
STRAIN(XYZ), PLASTIC_STRAIN(XYZ),
YIELD_FUNCTION

Small strains, viscoelastic: STRESS(XYZ), STRAIN(XYZ),
THERMAL_STRAIN(XYZ), ELEMENT_TEMPERATURE

Large strains, viscoelastic: STRESS (XYZ) ,
DEFORMATION_GRADIENT (XYZ) ,
THERMAL_STRAIN (XYZ) , ELEMENT_TEMPERATURE

Small strains: creep-irradiation: STRESS (XYZ) ,
STRAIN (XYZ) , PLASTIC_STRAIN (XYZ) ,
CREEP_STRAIN (XYZ) , THERMAL_STRAIN (XYZ) ,
ELEMENT_TEMPERATURE ,
ACCUMULATED_EFFECTIVE_PLASTIC_STRAIN ,
EFFECTIVE_STRESS , YIELD_STRESS ,
EFFECTIVE_CREEP_STRAIN ,
IRRADIATION_STRAIN (XYZ) .

Large strains: creep-irradiation: STRESS (XYZ) ,
DEFORMATION_GRADIENT (XYZ) ,
THERMAL_STRAIN (XYZ) , ELEMENT_TEMPERATURE ,
ACCUMULATED_EFFECTIVE_PLASTIC_STRAIN ,
EFFECTIVE_STRESS , YIELD_STRESS ,
EFFECTIVE_CREEP_STRAIN ,
IRRADIATION_STRAIN (XYZ) .

In the above lists,

STRESS (XYZ) = STRESS-YY, STRESS-ZZ, STRESS-
YZ, STRESS-XX

STRESS (ABC) = STRESS-AA, STRESS-BB, STRESS-
AB, STRESS-CC

with similar definitions for the other abbreviations used above. But
the variable DEFORMATION_GRADIENT (XYZ) is interpreted as
follows:

DEFORMATION_GRADIENT (XYZ) =
DEFORMATION_GRADIENT-YY,
DEFORMATION_GRADIENT-ZZ,
DEFORMATION_GRADIENT-XX,
DEFORMATION_GRADIENT-YZ,
DEFORMATION_GRADIENT-ZY

Also note that you can request stretches instead of deformation
gradients, and in this case STRETCH (XYZ) replaces
DEFORMATION_GRADIENT (XYZ) in the above lists.

See Section 13.1.1 for the definitions of those variables that are not self-explanatory.

ref. KJB
Section 4.2.1,
6.6.3

- In ADINA, the stresses are calculated using the strains at the point of interest. Hence, they are not spatially extrapolated or smoothed. However, the AUI can be employed to calculate smoothed stresses (see Section 13.1.3).
- You can also request that strain energy densities be output along with the stresses.

Variable categories

The variables are grouped into the following categories.

Stress (no saving, basic saving, all saving), default all saving

Strain (no saving, all saving), default no saving

Inelastic (no saving, basic saving, all saving), default basic saving

Thermal (no saving, basic saving, all saving), default basic saving

Energy (no saving, all saving), default no saving

Electromagnetic (no saving, all saving), default no saving

User-coded variables (no saving, all saving), default no saving

Misc (no saving, all saving), default no saving

Notice that some categories allow "basic" saving, namely only those results that are commonly used in post-processing are saved.

- The intent of the variable category feature is to decrease the number of results written to the porthole file. In particular notice that not all results are saved to the porthole file by default.
- The RESULTS-ELEMENT command is used to control the

saving of each variable category. To set the saving for a given 2D solid element group in the model, use the command RESULTS-ELEMENT ... GROUP=(group number). To set the saving for all remaining element groups in the model, use the command RESULTS-ELEMENT ... GROUP=0.

- The category saving feature can be turned off using the command PORTHOLE ... RESULTS-ELEMENT=NO. When PORTHOLE ... RESULTS-ELEMENT=NO, all results are saved at each element integration point. The default is PORTHOLE ... RESULTS-ELEMENT=YES, so that the category saving feature is used by default.
- In order to save all results, either set each category in RESULTS-ELEMENT to "all", or use the command PORTHOLE ... RESULTS-ELEMENT=NO.

Nodal forces: The nodal forces which correspond to the element stresses can also be requested in ADINA. This force vector is calculated in a linear analysis using

$$\mathbf{F} = \int_V \mathbf{B}^T \boldsymbol{\tau} dV$$

where \mathbf{B} is the element strain displacement matrix, and $\boldsymbol{\tau}$ is the stress vector. The integration is performed over the volume of the element; see Example 5.11 in ref. KJB, pp. 358-359.

The nodal forces are accessible in the AUI using the variable names NODAL_FORCE-X, NODAL_FORCE-Y, NODAL_FORCE-Z.

- The same relation is used for the element force calculation in a nonlinear analysis, except that updated quantities are used in the integration.

2.2.6 Recommendations on use of elements

- The 9-node element is usually most effective (except in explicit dynamic analysis).

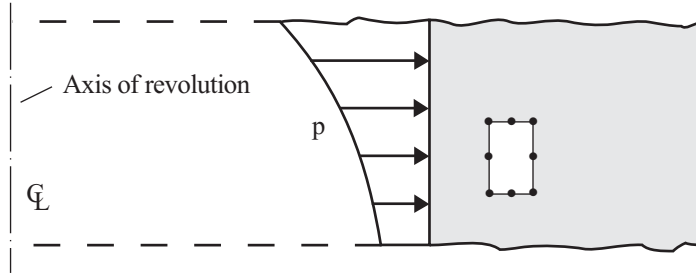
- The 8-node and 9-node elements can be employed for analysis of thick structures and solids, such as (see Fig. 2.2-7):

- ▶ Thick cylinders (axisymmetric idealization)
- ▶ Dams (plane strain idealization)
- ▶ Membrane sheets (plane stress idealization)
- ▶ Turbine blades (generalized plane strain idealization)

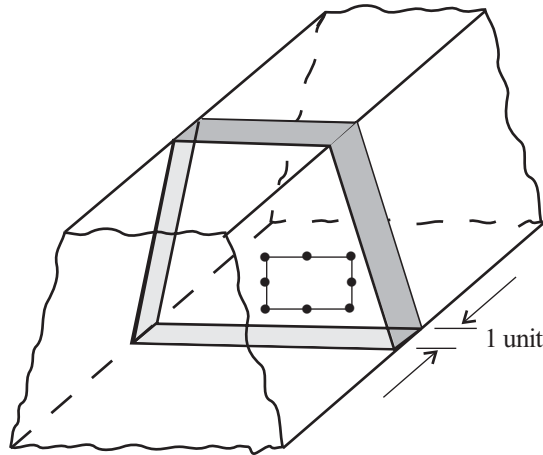
and for thin structures, such as (see Fig. 2.2-8):

- ▶ Plates and shells (axisymmetric idealization)
- ▶ Long plates (plane strain idealization)
- ▶ Beams (plane stress idealization)

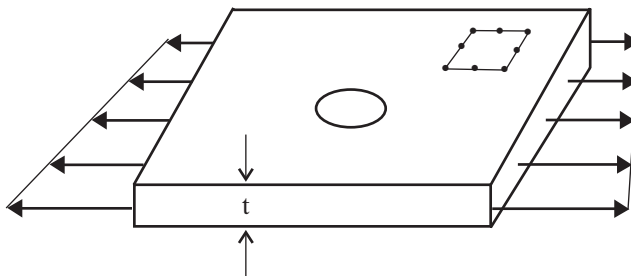
- The 8- and 9-node elements are usually most effective if the element is rectangular (non-distorted).
- The 4-node rectangular and 3-node triangular elements should only be used in analyses when bending effects are not significant. If the 4-node rectangular elements are used when bending effects are significant, consider the use of the incompatible modes element (see above).
- When the structure to be modeled is axisymmetric and has a dimension which is very small compared with the others, e.g., thin plates and shells, the use of axisymmetric shell elements is more effective (see Section 2.5).
- Geometrically nonlinear incompatible modes elements with large aspect ratio should not be used, because spurious modes may be present in the finite element solution.



a) Thick cylinder (axisymmetric idealization)

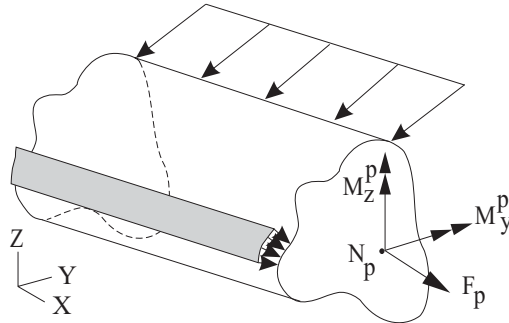


b) Long dam (plane strain idealization)



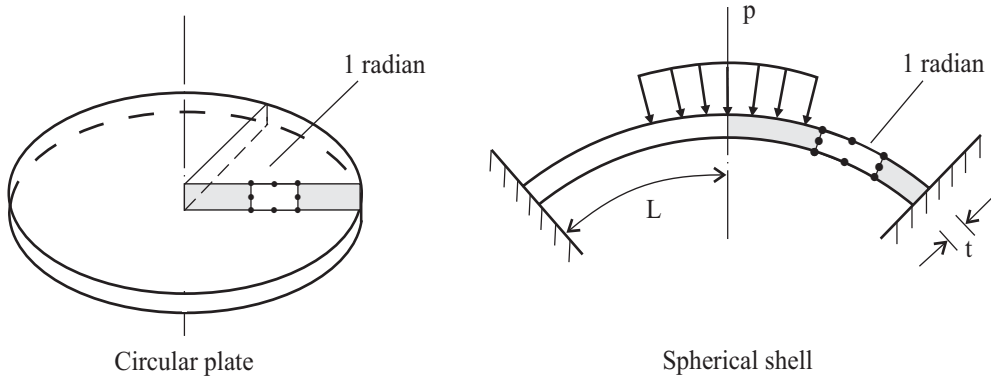
c) Sheet in membrane action (plane stress idealization)

Figure 2.2-7: Use of 2-D solid element for thick structures and solids

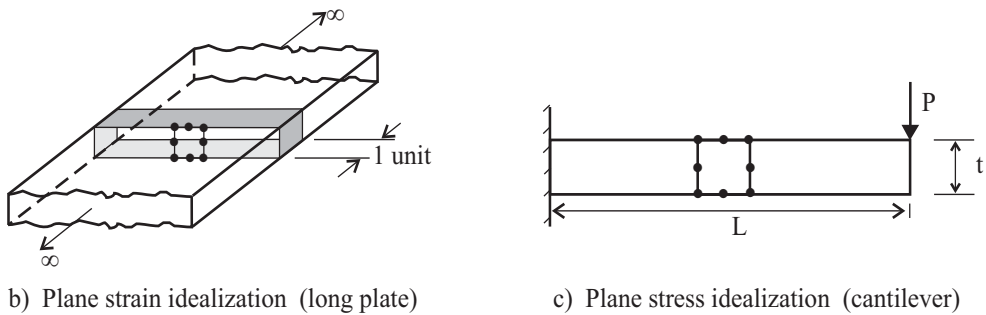


d) Turbine blade (generalized plane strain idealization)

Figure 2.2-7: (continued)



a) Axisymmetric idealizations



b) Plane strain idealization (long plate)

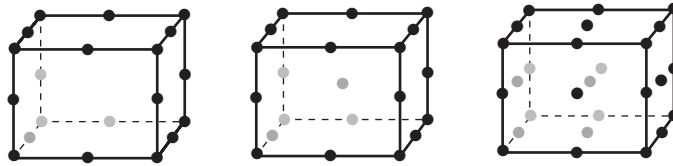
c) Plane stress idealization (cantilever)

Figure 2.2-8: Use of 2-D solid element for thin structures

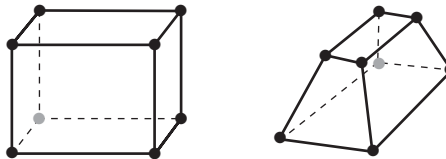
2.3 Three-dimensional solid elements

2.3.1 General considerations

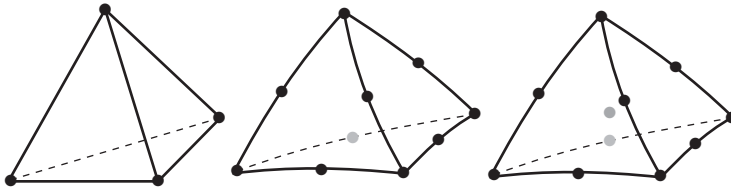
- The three-dimensional (3-D) solid element is a variable 4- to 20-node or a 21- or 27-node isoparametric element applicable to general 3-D analysis. Some typical 3-D solid elements are shown in Fig. 2.3-1.



(a) 20-, 21- & 27-node elements



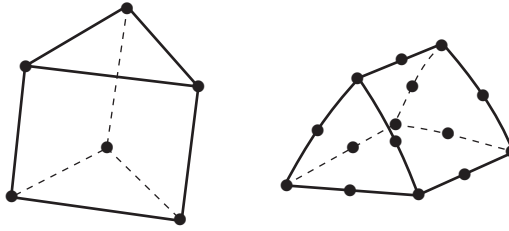
(b) 8-node elements



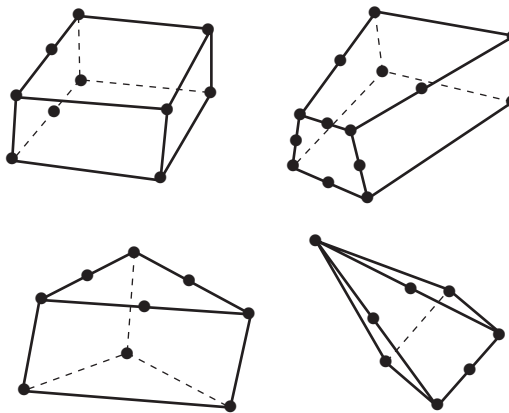
(c) 4-, 10- and 11-node tetrahedral elements

Figure 2.3-1: Some 3-D solid elements

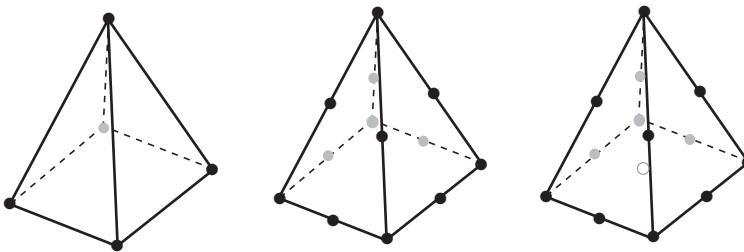
- The 3-D solid element should be employed in analyses in which the three-dimensional state of stress (or strain) is required or in which special stress/strain conditions, such as those given in Section 2.2, do not exist (see Figs. 2.3-2 and 2.3-3).



(d) 6- and 15- node prisms



(e) Elements for transition zones



(f) 5-, 13-, and 14-pyramid elements

(f) 5, 13-node and 14-node pyramid elements

Figure 2.3-1 (continued)

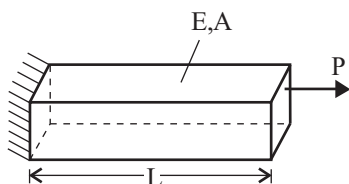
ref. KJB • The elements usually used are isoparametric displacement-based
Section 5.3

finite elements, and the formulation of the elements used in ADINA is described in ref. KJB, Ch. 5.

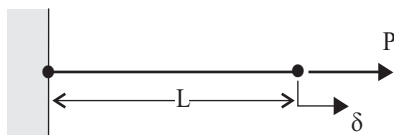
- The basic finite element assumptions are (see Fig. 2.3-4):

for the coordinates

$$x = \sum_{i=1}^q h_i x_i \quad y = \sum_{i=1}^q h_i y_i \quad z = \sum_{i=1}^q h_i z_i$$

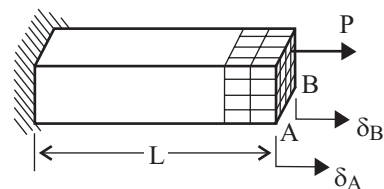


(a) Physical problem considered



Plane sections remain plane. δ is constant for the entire section.

(b) 1-D model



Plane sections do not remain plane.

$$\delta_A \neq \delta_B$$

(c) 3-D model

Figure 2.3-2: Illustration of use of 3-D elements

and for the displacements

$$u = \sum_{i=1}^q h_i u_i \quad v = \sum_{i=1}^q h_i v_i \quad w = \sum_{i=1}^q h_i w_i$$

where

$h_i(r, s, t)$ = interpolation function corresponding to node i

r, s, t = isoparametric coordinates

q = number of element nodes, $4 \leq q \leq 27$

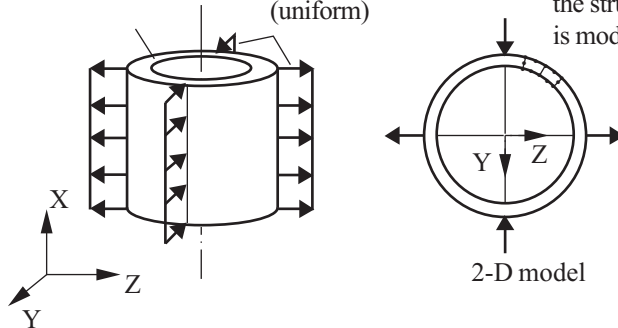
x_i, y_i, z_i = nodal point coordinates

u_i, v_i, w_i = nodal point displacements

No axial displacement
should be possible

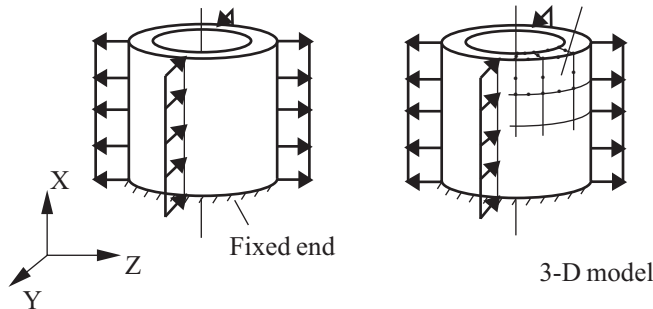
Line loads
(uniform)

One quarter of
the structure
is modeled



a) Plane strain 2-D analysis applicable

One quarter of the
structure is modeled



b) Three-dimensional model is used

Figure 2.3-3: Examples of 3-D solid modeling versus 2-D solid modeling

- In addition to the displacement-based elements, special mixed-interpolated elements are also available, in which the displacements and pressure are interpolated. These elements are effective and should be preferred in the analysis of incompressible media and inelastic materials (specifically for materials in which Poisson's ratio is close to 0.5, for rubber-like materials and for elasto-plastic materials).
- The mixed formulation is the default setting for the materials in which either the entire response is incompressible (for example, the rubber material models), or in which the inelastic response is incompressible (for example, the plastic and creep material models).
- Table 2.3-1 shows the default number of pressure degrees of freedom for each 3-D element type. For more details on the mixed interpolation of pressure and displacement degrees of freedom for 3-D solids, see Section 4.4.3, p. 276, and Tables 4.6 and 4.7, pp.292-295 in ref. KJB.
- The default number of pressure degrees of freedom shown in Table 2.3-1 is optimal for most applications. However, it can be modified if needed. In some problems where rubber is subjected to severe compression, numerical instability can be avoided by using 8 pressure degrees of freedom for the 20, 21 and 27-node bricks.
- For brick elements, the following choices for the number of pressure degrees of freedom are valid; 1 for a constant pressure, 4 for additional linear pressure terms, 7 for additional bilinear pressure terms (involving r_s , r_t , s_t), 8 for an additional tri-linear pressure term (involving rst), and 20 for additional quadratic pressure terms (involving r^2 , s^2 , t^2). For tetrahedral elements, the following choices for the number of pressure degrees of freedom are valid; 1 for a constant pressure, 4 for additional linear pressure terms, 10 for additional quadratic terms, 11 for an additional “bubble” term at the center of the element and 15 for additional cubic terms.
- In general, using too many pressure degrees of freedom leads to volumetric locking, while using too few pressure degrees of freedom leads to spurious zero-energy modes. See Sections 4.4.2

and 4.5 in ref. KJB for more details.

Table 2.3-1: Mixed u/p formulations available for 3-D solid elements

3-D solid element	Number of default pressure DOFs
4-node tetrahedron	1 [*]
10-node tetrahedron (10/4)	4
11-node tetrahedron (11/4)	4
6-node prism	1
15-node prism	4
21-node prism	4
8-node brick (8/1)	1
20-node brick (20/4)	4
21-node brick	4
27-node brick (27/4)	4
5-node pyramid	1
13-node pyramid	4
14-node pyramid	4

^{*} To use this element, the user must use the displacement-based element with 1-point integration.

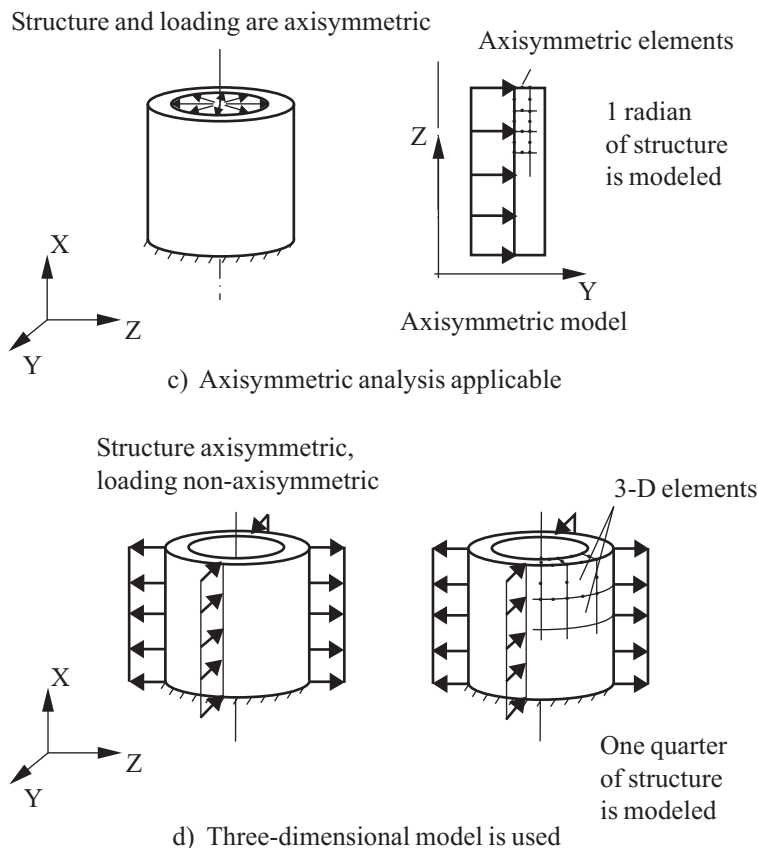


Figure 2.3-3: (continued)

- In addition to the displacement-based and mixed-interpolated elements, ADINA also includes the possibility of including incompatible modes in the formulation of the 8-node element. The addition of the incompatible modes increases the flexibility of the element, especially in bending situations. This element is analogous to the 4-node 2-D solid element with incompatible modes discussed in Section 2.2.1, see the comments in that section for theoretical considerations.

The incompatible modes feature cannot be used in conjunction with the mixed-interpolation formulation.

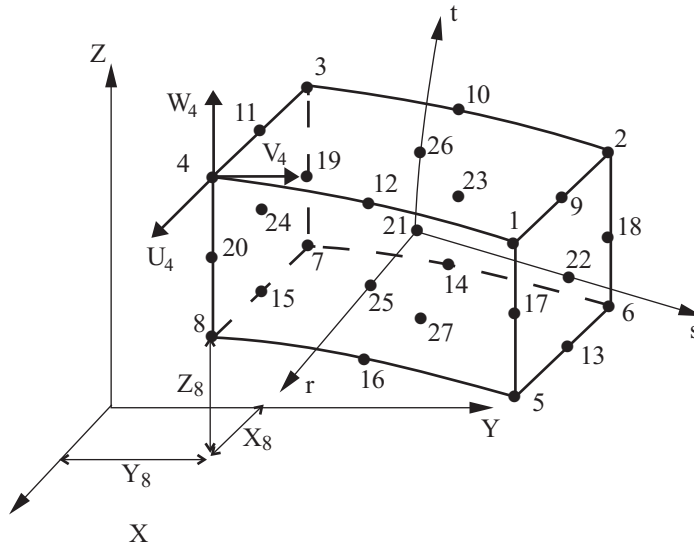


Figure 2.3-4: Conventions used for the nodal coordinates and displacements of the 3-D solid element

The incompatible modes feature is only available for the 8-node element, and elements degenerated from the 8-node element, except for the 4-node tetrahedral element.

- The elements can be used with 4 to 20 or with 21 or 27 nodes (tetrahedra, pyramids or prisms are derived from the degeneration of the 4 to 20-node rectangular elements, see Example 5.16 on pp. 366-367, ref. KJB). The interpolation functions for $q \leq 20$ are shown in Fig. 5.5, ref. KJB, p. 345.
- Degenerated elements such as prisms, pyramids or tetrahedra are formed in ADINA by assigning the same global node to the local element nodes located along the same side or on the same face.

The triangular prism as a degenerated 20-node element (see Fig. 2.3-1(d)) can be used in ADINA in three different ways:

- As a 15-node triangular prism with the strain singularity $\frac{1}{\sqrt{r}}$. This is an extension of the techniques described on pp.

368-376 of ref. KJB (see also Chapter 10);

- As a 20-node triangular prism with the strain singularity $\frac{1}{r}$

(see also Chapter 10);

- As a spatially isotropic triangular prism where corrections are applied to the interpolation functions of the collapsed 20-node element.

The 10-node tetrahedron (see Fig. 2.3-1(c)) is obtained by collapsing nodes and sides of rectangular elements. A spatially isotropic 10-node tetrahedron is available in ADINA. Similarly, an 11-node spatially isotropic tetrahedron is available in ADINA.

The 13-node pyramid and the 14-node pyramid (see Fig. 2.3-1(f)) are degenerate solid elements from 20-node and 21-node bricks, respectively. They can be used as spatially isotropic elements.

The 4-node tetrahedron (see Fig. 2.3-1(c)) is obtained by collapsing nodes and sides of the 8-node rectangular element. This element exhibits constant strain conditions.

The elements in Fig. 2.3-1(e) are not spatially isotropic and should usually only be employed in transition regions.

- Linear and nonlinear fracture mechanics analysis of stationary cracks can be performed with three-dimensional elements (see Chapter 10).

2.3.2 Material models and nonlinear formulations

- The 3-D elements can be used with the following material models: **elastic-isotropic, elastic-orthotropic, nonlinear-elastic, plastic-bilinear, plastic-multilinear, Drucker-Prager, Mroz-bilinear, plastic-orthotropic, thermo-isotropic, thermo-orthotropic, thermo-plastic, piezoelectric, creep, plastic-creep, multilinear-plastic-creep, concrete, curve-description, Ogden, Mooney-Rivlin, Arruda-Boyce, hyper-foam, Sussman-Bathe, creep-variable, plastic-creep-variable, multilinear-plastic-creep-variable, Gurson-plastic, Cam-clay, Mohr-Coulomb, viscoelastic, irradiation creep, gasket, shape-memory alloy, plastic-cyclic, user-coded.**

- The mixed interpolation formulation should be used (and is the default) whenever using the plastic-bilinear, plastic-multilinear, Mroz- bilinear, plastic-orthotropic, thermo-plastic, creep, plastic-creep, multilinear-plastic-creep, Ogden, Mooney-Rivlin, Arruda-Boyce, Sussman-Bathe, creep-variable, plastic-creep-variable, multilinear-plastic-creep-variable, plastic-cyclic or viscoelastic material models.

In addition the mixed interpolation formulation should be used (but is not the default) when using the elastic-isotropic material and the Poisson's ratio is close to 0.5.

- The 3-D elements can be used with a **small displacement/small strain, large displacement/small strain** or a **large displacement/large strain** formulation.

The small displacement/small strain and large displacement/small strain formulations can be used with any material model, except for the Ogden, Mooney-Rivlin, Arruda-Boyce, hyper-foam and Sussman-Bathe material models. The use of a linear material with the small displacement/small strain formulation corresponds to a linear formulation, and the use of a nonlinear material with the small displacement/small strain formulation corresponds to a materially-nonlinear-only formulation. The program uses the TL (total Lagrangian) formulation when you choose a large displacement/small strain formulation.

The large displacement/large strain formulations can be used with the plastic-bilinear, plastic-multilinear, Mroz-bilinear, plastic-orthotropic, thermo-plastic, creep, plastic-creep, multilinear-plastic-creep, creep-variable, plastic-creep-variable, multilinear-plastic-creep-variable, viscoelastic, plastic-cyclic and user-coded material models. The program uses either a ULH or ULJ formulation when you choose a large displacement/large strain formulation with these material models.

A large displacement/large strain formulation is used with the Ogden, Mooney-Rivlin, Arruda-Boyce, hyper-foam and Sussman-Bathe material models. The program uses a TL (total Lagrangian) formulation in this case.

*ref. KJB
Sections 6.2
and 6.3.5*

- The basic continuum mechanics formulations are described in ref. KJB, pp. 497-568. The finite element discretization is summarized in Table 6.6, p. 555, ref. KJB.

ref. KJB
Section 6.8.1

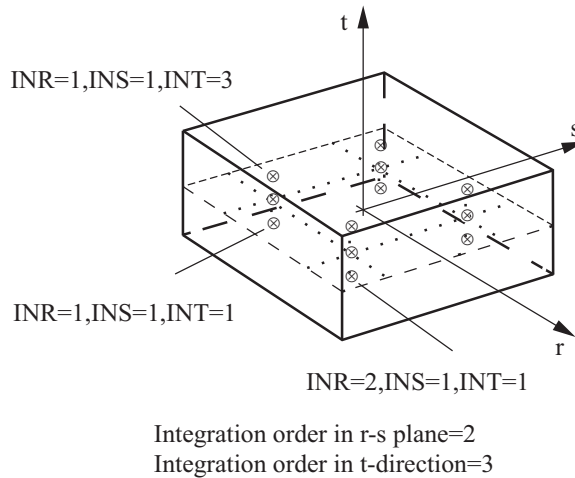
- Note that all these formulations can be used in one finite element mesh. If the elements are initially compatible, then they will remain compatible throughout the analysis.

2.3.3 Numerical integration

Hexahedral (brick) elements, including collapsed hexahedral elements

ref. KJB
Sections 5.5.3,
5.5.4 and 5.5.5

- For the calculation of element matrices, Gauss numerical integration is used. The same integration order (RSINT) is always assigned to the r- and s-directions, and can be from 2 to 6. The integration order in the t-direction (TINT) can, however, be assigned independently, and can also be from 2 to 6. The default Gauss integration orders are $2 \times 2 \times 2$ for the 8-node (cube or prism) elements and $3 \times 3 \times 3$ otherwise, except for tetrahedra.
- Except for the 4-node, 10-node and 11-node tetrahedra, the convention for the integration point numbering used in the stress output is as follows: The first integration point is the point with the most negative location in r, s and t. The next integration points are located by increasing t (and label INT) successively up to its maximum positive value, then increasing s (and label INS) one position towards positive and varying t again from its maximum negative to its maximum positive values, and so on (Fig. 2.3-5).



a) All elements except tetrahedra

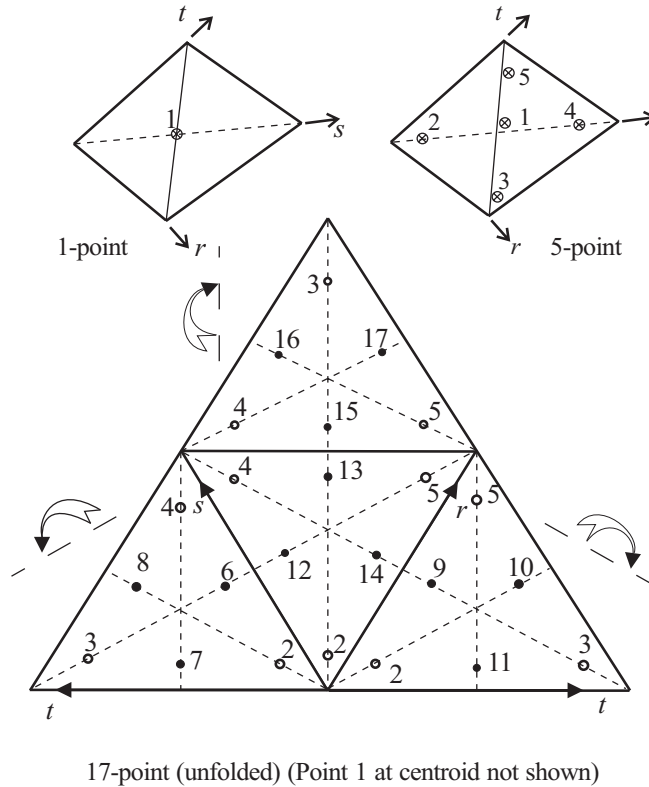
Figure 2.3-5: Example of integration point labeling for 3-D solid elements

4, 10 and 11-node tetrahedral elements

- Tetrahedral elements are spatially isotropic with respect to integration point locations and interpolation functions (see Figure 2.3-5(b)). For the 4-node tetrahedral element, 1-point Gauss integration is used. For the 10-node and 11-node tetrahedral elements, TETINT Gauss integration points are used, where TETINT can be 1, 4, 5, 17.

TETINT=DEFAULT is also allowed, and is the default. When TETINT=DEFAULT, the number of tetrahedral integration points is set based on the maximum number of nodes for any element in the element group:

Max number of nodes in any brick element	Max number of nodes in any tetrahedral element	TETINT
≤ 8	≤ 4	1
≤ 20	≤ 10	5
≤ 27	≤ 11	17



b) Tetrahedral elements

Figure 2.3-5: (continued)

General Comments

- Note that in geometrically nonlinear analysis, the spatial positions of the Gauss integration points change continuously as the element undergoes deformations, but throughout the response the same material particles are at the integration points.
- The order of numerical integration in large displacement elastic analysis is usually best chosen to be equal to the order appropriate in linear elastic analysis. Hence the default values are usually appropriate. However, in certain inelastic analyses, a higher order should be used (see Fig. 6.25, p. 638, ref. KJB).

ref. KJB
Section 6.8.4

2.3.4 Mass matrices

- The consistent mass matrix is always calculated using $3 \times 3 \times 3$ Gauss integration except for the tetrahedral 4-node, 10-node and 11-node elements which use a 17-point Gauss integration.
- The lumped mass matrix of an element is formed by dividing the element's mass M equally among each of its n nodal points. Hence the mass assigned to each node is M/n . No special distributory concepts are employed to distinguish between corner and midside nodes, or to account for element distortion.
- Note that n is the number of distinct non-repeated nodes in the element. Hence, when an element side or face is collapsed to a single node, the total mass of the element is divided among the unique nodes in the element.
- For the following element shapes, the lumped mass is divided among the collapsed nodes: 4-node tetrahedron, 10-node tetrahedron, 11-node tetrahedron, 5-node pyramid, 13-node pyramid, 14-node pyramid, 6-node prism, 15-node prism, 21-node prism. For example, in the 4-node tetrahedron, each of the four nodes has one quarter of the total mass.

2.3.5 Element output

- You can request that ADINA either print or save stresses or forces. However ADINA cannot output both stresses and forces in the same run.

Stresses

Each element outputs the following information to the porthole file, based on the material model. This information is accessible in the AUI using the given variable names.

Notice that for the orthotropic and piezoelectric material models, you can request that the stresses and strains be saved in the material coordinate system.

Elastic-isotropic, elastic-orthotropic (results saved in global system): STRESS (XYZ) , STRAIN (XYZ)

Elastic-isotropic with thermal effects: STRESS (XYZ) ,
STRAIN (XYZ) , THERMAL_STRAIN,
ELEMENT_TEMPERATURE

Orthotropic (results saved in material system): STRESS (ABC) ,
STRAIN (ABC)

Thermo-isotropic, thermo-orthotropic (results saved in global system): STRESS (XYZ) , STRAIN (XYZ) ,
THERMAL_STRAIN (XYZ) , ELEMENT_TEMPERATURE

Piezoelectric (results saved in global system): STRESS(XYZ),
STRAIN(XYZ), ELECTRIC_FIELD-X, ELECTRIC_FIELD-Y,
ELECTRIC_FIELD-Z, ELECTRIC_DISPLACEMENT_FIELD-X,
ELECTRIC_DISPLACEMENT_FIELD-Y,
ELECTRIC_DISPLACEMENT_FIELD-Z.

Piezoelectric (results saved in material system): STRESS(ABC),
STRAIN(ABC), ELECTRIC_FIELD-A, ELECTRIC_FIELD-B,
ELECTRIC_FIELD-C, ELECTRIC_DISPLACEMENT_FIELD-A,
ELECTRIC_DISPLACEMENT_FIELD-B,
ELECTRIC_DISPLACEMENT_FIELD-C.

Curve description: CRACK_FLAG, STRESS (XYZ) ,
STRAIN (XYZ) , FE_SIGMA-P1, FE_SIGMA-P2,
FE_SIGMA-P3, GRAVITY_IN-SITU_PRESSURE,
VOLUMETRIC_STRAIN, FE_SIGMA-P1_DIRECTION-X,
FE_SIGMA-P1_DIRECTION-Y,
FE_SIGMA-P1_DIRECTION-Z,
FE_SIGMA-P2_DIRECTION-X,
FE_SIGMA-P2_DIRECTION-Y,
FE_SIGMA-P2_DIRECTION-Z,
FE_SIGMA-P3_DIRECTION-X,
FE_SIGMA-P3_DIRECTION-Y,
FE_SIGMA-P3_DIRECTION-Z

Concrete: CRACK_FLAG, STRESS (XYZ) , STRAIN (XYZ) ,
FE_SIGMA-P1, FE_SIGMA-P2, FE_SIGMA-P3,
FAILURE_STRESS, ELEMENT_TEMPERATURE, FE_SIGMA-
P1_DIRECTION-X,

```
FE_SIGMA-P1_DIRECTION-Y,  
FE_SIGMA-P1_DIRECTION-Z,  
FE_SIGMA-P2_DIRECTION-X,  
FE_SIGMA-P2_DIRECTION-Y,  
FE_SIGMA-P2_DIRECTION-Z,  
FE_SIGMA-P3_DIRECTION-X,  
FE_SIGMA-P3_DIRECTION-Y,  
FE_SIGMA-P3_DIRECTION-Z, THERMAL_STRAIN(XYZ)
```

Small strains: Plastic-bilinear, plastic-multilinear, Mroz-bilinear:

```
PLASTIC_FLAG, STRESS(XYZ), STRAIN(XYZ),  
PLASTIC_STRAIN(XYZ), FE_EFFECTIVE_STRESS,  
YIELD_STRESS,  
ACCUMULATED_EFFECTIVE_PLASTIC_STRAIN,  
THERMAL_STRAIN(XYZ), ELEMENT_TEMPERATURE
```

Large strains: Plastic-bilinear, plastic-multilinear, Mroz-bilinear:

```
PLASTIC_FLAG, STRESS(XYZ),  
DEFORMATION_GRADIENT(XYZ),  
FE_EFFECTIVE_STRESS, YIELD_STRESS,  
ACCUMULATED_EFFECTIVE_PLASTIC_STRAIN,  
THERMAL_STRAIN(XYZ), ELEMENT_TEMPERATURE
```

```
SMA: EFFECTIVE_STRESS, STRESS-XX, STRESS-XY,  
STRESS-XZ, STRESS-YY, STRESS-YZ, STRESS-ZZ,  
ACCUM_EFF_TRANSF_STRAIN, AUSTENITE_FRACTION,  
DETWINNED_MARTENSITE_FRACTION, STRAIN-XX,  
STRAIN-XY, STRAIN-XZ, STRAIN-YY, STRAIN-YZ,  
STRAIN-ZZ, THERMAL_STRAIN-XX, THERMAL_STRAIN-  
YY, THERMAL_STRAIN-ZZ, TRANSFORMATION_STRAIN-  
XX, TRANSFORMATION_STRAIN-XY,  
TRANSFORMATION_STRAIN-XZ,  
TRANSFORMATION_STRAIN-YY,  
TRANSFORMATION_STRAIN-YZ,  
TRANSFORMATION_STRAIN-ZZ,  
TWINNED_MARTENSITE_FRACTION, SMA_FLAG
```

Small strains: Anand Model

```
STRESS(XYZ), STRAIN(XYZ),  
VISCOPLASTIC_STRAIN(XYZ),  
FE_EFFECTIVE_STRESS,  
ACCUMULATED_EFFECTIVE_VISCOPLASTIC_STRAIN,  
EFFECTIVE_VISCOPLASTIC_STRAIN_RATE,  
THERMAL_STRAIN(XYZ), ELEMENT_TEMPERATURE
```


Large strains: Anand Model

STRESS(XYZ), DEFORMATION_GRADIENT(XYZ),
FE_EFFECTIVE_STRESS,
ACCUMULATED_EFFECTIVE_VISCOPLASTIC_STRAIN,
EFFECTIVE_VISCOPLASTIC_STRAIN_RATE,
THERMAL_STRAIN(XYZ), ELEMENT_TEMPERATURE

Plastic-orthotropic (results saved in global system):

PLASTIC_FLAG, STRESS(XYZ), STRAIN(XYZ),
PLASTIC_STRAIN(XYZ), HILL_EFFECTIVE_STRESS,
YIELD_STRESS,
ACCUMULATED_EFFECTIVE_PLASTIC_STRAIN,
THERMAL_STRAIN(XYZ), ELEMENT_TEMPERATURE

Plastic-orthotropic (results saved in material system):

PLASTIC_FLAG, STRESS(ABC), STRAIN(ABC),
PLASTIC_STRAIN(ABC), HILL_EFFECTIVE_STRESS,
YIELD_STRESS,
ACCUMULATED_EFFECTIVE_PLASTIC_STRAIN,
THERMAL_STRAIN(ABC), ELEMENT_TEMPERATURE

Small strains: Drucker-Prager: PLASTIC_FLAG,

PLASTIC_FLAG-2, STRESS(XYZ), STRAIN(XYZ),
PLASTIC_STRAIN(XYZ), CAP_LOCATION,
YIELD_FUNCTION

Large strains: Drucker-Prager: PLASTIC_FLAG,

PLASTIC_FLAG-2, STRESS(XYZ),
DEFORMATION_GRADIENT(XYZ), CAP_LOCATION,
YIELD_FUNCTION

Small strains: Thermo-plastic, creep, plastic-creep, multilinear-
plastic-creep, creep-variable, plastic-creep-variable, multilinear-
plastic-creep-variable: PLASTIC_FLAG,

NUMBER_OF_SUBINCREMENTS, STRESS(XYZ),
STRAIN(XYZ), PLASTIC_STRAIN(XYZ),
CREEP_STRAIN(XYZ), THERMAL_STRAIN(XYZ),
ELEMENT_TEMPERATURE,
ACCUMULATED_EFFECTIVE_PLASTIC_STRAIN,
FE_EFFECTIVE_STRESS, YIELD_STRESS,
EFFECTIVE_CREEP_STRAIN

Large strains: Thermo-plastic, creep, plastic-creep, multilinear-plastic-creep, creep-variable, plastic-creep-variable, multilinear-plastic-creep-variable: PLASTIC_FLAG,
NUMBER_OF_SUBINCREMENTS, STRESS(XYZ),
DEFORMATION_GRADIENT(XYZ),
THERMAL_STRAIN(XYZ), ELEMENT_TEMPERATURE,
ACCUMULATED_EFFECTIVE_PLASTIC_STRAIN,
FE_EFFECTIVE_STRESS, YIELD_STRESS,
EFFECTIVE_CREEP_STRAIN

Mooney-Rivlin, Ogden, Arruda-Boyce, hyper-foam (strains saved): STRESS(XYZ), STRAIN(XYZ)

Mooney-Rivlin, Ogden, Arruda-Boyce, hyper-foam (deformation gradients saved): STRESS(XYZ),
DEFORMATION_GRADIENT(XYZ)

Small strains: User-supplied: STRESS(XYZ), STRAIN(XYZ),
USER_VARIABLE_I

Large strains: User-supplied: STRESS(XYZ),
DEFORMATION_GRADIENT(XYZ), USER_VARIABLE_I

Thermo-orthotropic (results saved in material system):
STRESS(ABC), STRAIN(ABC),
THERMAL_STRAIN(ABC), FE_EFFECTIVE_STRESS,
ELEMENT_TEMPERATURE

Small strains: Gurson plastic: PLASTIC_FLAG,
STRESS(XYZ), STRAIN(XYZ),
PLASTIC_STRAIN(XYZ), FE_EFFECTIVE_STRESS,
YIELD_STRESS,
ACCUMULATED_EFFECTIVE_PLASTIC_STRAIN,
THERMAL_STRAIN(XYZ), ELEMENT_TEMPERATURE,
VOID_VOLUME_FRACTION

Large strains: Gurson plastic: PLASTIC_FLAG,
STRESS(XYZ), DEFORMATION_GRADIENT(XYZ),
FE_EFFECTIVE_STRESS, YIELD_STRESS,
ACCUMULATED_EFFECTIVE_PLASTIC_STRAIN,
THERMAL_STRAIN(XYZ), ELEMENT_TEMPERATURE,
VOID_VOLUME_FRACTION

Small strains: Cam-clay: PLASTIC_FLAG, STRESS(XYZ),
STRAIN(XYZ), PLASTIC_STRAIN(XYZ),
YIELD_SURFACE_DIAMETER_P, YIELD_FUNCTION,
MEAN_STRESS, DISTORTIONAL_STRESS,
VOLUMETRIC_STRAIN, VOID_RATIO,
EFFECTIVE_STRESS_RATIO, SPECIFIC_VOLUME

Large strains: Cam-clay: PLASTIC_FLAG, STRESS(XYZ),
DEFORMATION_GRADIENT(XYZ),
YIELD_SURFACE_DIAMETER_P, YIELD_FUNCTION,
MEAN_STRESS, DISTORTIONAL_STRESS,
VOLUMETRIC_STRAIN, VOID_RATIO,
EFFECTIVE_STRESS_RATIO, SPECIFIC_VOLUME

Mohr-Coulomb: PLASTIC_FLAG, STRESS(XYZ),
STRAIN(XYZ), PLASTIC_STRAIN(XYZ),
YIELD_FUNCTION

Small strains: viscoelastic: STRESS(XYZ), STRAIN(XYZ),
THERMAL_STRAIN(XYZ), ELEMENT_TEMPERATURE

Large strains: viscoelastic: STRESS(XYZ),
DEFORMATION_GRADIENT(XYZ),
THERMAL_STRAIN(XYZ), ELEMENT_TEMPERATURE

Small strains: creep-irradiation: STRESS(XYZ),
STRAIN(XYZ), PLASTIC_STRAIN(XYZ),
CREEP_STRAIN(XYZ), THERMAL_STRAIN(XYZ),
ELEMENT_TEMPERATURE,
ACCUMULATED_EFFECTIVE_PLASTIC_STRAIN,
EFFECTIVE_STRESS, YIELD_STRESS,
EFFECTIVE_CREEP_STRAIN,
IRRADIATION_STRAIN(XYZ).

Large strains: creep-irradiation: STRESS(XYZ),
DEFORMATION_GRADIENT(XYZ),
THERMAL_STRAIN(XYZ), ELEMENT_TEMPERATURE,
ACCUMULATED_EFFECTIVE_PLASTIC_STRAIN,
EFFECTIVE_STRESS, YIELD_STRESS,
EFFECTIVE_CREEP_STRAIN,
IRRADIATION_STRAIN(XYZ).

In the above lists,

```
STRESS (XYZ) = STRESS-XX, STRESS-YY, STRESS-  
ZZ, STRESS-XY, STRESS-XZ, STRESS-YZ
```

```
STRESS (ABC) = STRESS-AA, STRESS-BB, STRESS-  
CC, STRESS-AB, STRESS-AC, STRESS-BC
```

with similar definitions for the other abbreviations used above. But the variable `DEFORMATION_GRADIENT (XYZ)` is interpreted as follows:

```
DEFORMATION_GRADIENT (XYZ) =  
DEFORMATION_GRADIENT-XX,  
DEFORMATION_GRADIENT-XY,  
DEFORMATION_GRADIENT-XZ,  
DEFORMATION_GRADIENT-YX,  
DEFORMATION_GRADIENT-YY,  
DEFORMATION_GRADIENT-YZ,  
DEFORMATION_GRADIENT-ZX,  
DEFORMATION_GRADIENT-ZY,  
DEFORMATION_GRADIENT-ZZ,
```

Also note that you can request stretches instead of deformation gradients, and in this case `STRETCH (XYZ)` replaces `DEFORMATION_GRADIENT (XYZ)` in the above lists.

See Section 13.1.1 for the definitions of those variables that are not self-explanatory.

- You can also request that strain energy densities be output along with the stresses.

Variable categories

The variables are grouped into the following categories.

Stress (no saving, basic saving, all saving), default all saving

Strain (no saving, all saving), default no saving

Inelastic (no saving, basic saving, all saving), default basic

saving

Thermal (no saving, basic saving, all saving), default basic saving

Energy (no saving, all saving), default no saving

Electromagnetic (no saving, all saving), default no saving

User-coded variables (no saving, all saving), default no saving

Misc (no saving, all saving), default no saving

Notice that some categories allow "basic" saving, namely only those results that are commonly used in post-processing are saved.

- The intent of the variable category feature is to decrease the number of results written to the porthole file. In particular notice that not all results are saved to the porthole file by default.
- The RESULTS-ELEMENT command is used to control the saving of each variable category. To set the saving for a given 3D solid element group in the model, use the command RESULTS-ELEMENT ... GROUP=(group number). To set the saving for all remaining 3D solid element groups in the model, use the command RESULTS-ELEMENT ... GROUP=0.
- The category saving feature can be turned off using the command PORTHOLE ... RESULTS-ELEMENT=NO. When PORTHOLE ... RESULTS-ELEMENT=NO, all results are saved at each element integration point. The default is PORTHOLE ... RESULTS-ELEMENT=YES, so that the category saving feature is used by default.
- In order to save all results, either set each category in RESULTS-ELEMENT to "all", or use the command PORTHOLE ... RESULTS-ELEMENT=NO.
- In ADINA, the stresses are calculated using the strains at the point of interest. The stresses are not smoothed. The AUI can be employed to calculate smoothed stresses from the results output by ADINA.

ref. KJB
Section 4.2.1

Locations where results are saved

The results are saved either at the element integration points or at the element corner nodes. When the results are saved at the integration points, the stresses are calculated using the strains at the point of interest. Hence, they are not spatially extrapolated or smoothed. The AUI can be employed to calculate smoothed stresses from the results output by ADINA.

- When the results are saved at the element corner nodes, the number of points per element at which results are saved is significantly reduced. For example, if results are saved at the corner nodes of a 27-node brick element, there are 8 points per element, as opposed to 27 integration points; and if results are saved at the corner nodes of an 11-node tetrahedral element, there are 4 points per element, as opposed to 17 integration points.
- The RESULTS-ELEMENT command is used to control where the results are saved. The default is to save the results at the element integration points.

Nodal forces

The nodal forces that correspond to the element stresses can also be requested in ADINA. This force vector is calculated in a linear analysis using

$$\mathbf{F} = \int_V \mathbf{B}^T \boldsymbol{\tau} dV$$

where \mathbf{B} is the element strain-displacement matrix and $\boldsymbol{\tau}$ is the stress vector. The integration is performed over the volume of the element; see Example 5.11 in ref. KJB, pp. 358-359.

The nodal forces are accessible in the AUI using the variable names NODAL_FORCE-X, NODAL_FORCE-Y, NODAL_FORCE-Z.

- The same relation is used for the element force calculation in a nonlinear analysis, except that updated quantities are used in the

integration.

2.3.6 Recommendations on use of elements

- The 27-node element is the most accurate among all available elements. However, the use of this element can be costly.
- The 20-node element is usually the most effective (except in wave propagation analysis using the central difference method and a lumped mass idealization, see Chapter 7).
- The 20-node element can be employed for analysis of thick structures and solids; some examples are given in Fig. 2.3-6.
- The 20-node element is usually most effective if the element is rectangular (undistorted).
- The 11-node tetrahedral element should be used when tetrahedral meshing is used and a mixed interpolated formulation is required (as in incompressible analysis).

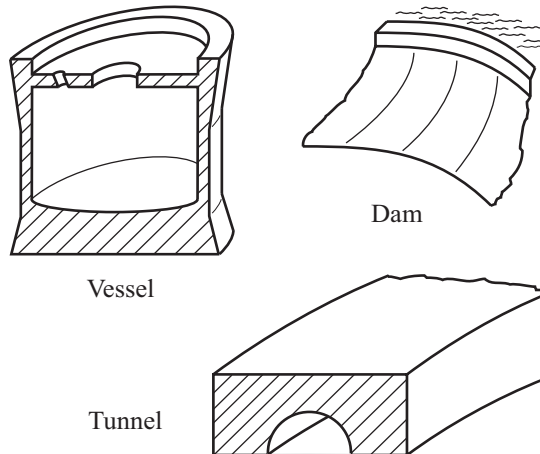


Figure 2.3-6: Some structures for which 3-D solid elements can be used

- The 8-node brick element and 4-node tetrahedral element

should only be used in analyses when bending effects are not significant. If the 8-node brick element must be used when bending effects are significant, consider the use of the incompatible modes element.

- Geometrically nonlinear incompatible modes elements with large aspect ratio should not be used, because spurious modes may be present in the finite element solution.
- When the structure to be modeled has a dimension which is extremely small compared with the others, e.g., thin plates and shells, the use of the 3-D solid element usually results in too stiff a model and a poor conditioning of the stiffness matrix. In this case, the use of the shell or the plate/shell elements (see Sections 2.7 and 2.6) is more effective.

2.4 Beam elements

- The beam element is a 2-node Hermitian beam with a constant cross-section. The element is initially straight.
- The beam element can be employed in the following analysis conditions:
 - ▶ Linear analysis, in which case the displacements, rotations and strains are infinitesimally small, and the material is linear elastic.
 - ▶ Materially nonlinear only analysis, in which case the displacements, rotations and strains are infinitesimally small, but the material is nonlinear.
 - ▶ Large displacement/large rotation analysis, in which case the displacements and rotations can be large, but the strains are small. The material can either be linear or nonlinear.
- The beam element can be used in statics, implicit dynamics, explicit dynamics and frequency analysis.
- The material behavior of the beam can be described using either

a cross-section shape and a material model, or a moment-curvature input.

- The beam element can optionally include warping degrees of freedom. This option is suitable when modeling thin-walled cross-sections, especially when the angle of twist per unit length is not constant along the beam. Throughout this section, the beam element without warping degrees of freedom is referred to as the standard beam element, and the beam element with warping degrees of freedom is referred to as the warping beam element.
- The available beam capabilities are summarized in Table 2.4.1.
- The beam element can optionally include a bolt feature. The bolt feature is fully described in Section 11.17.
- Throughout this section, the element formulations of the current ADINA system are described. It is, however, possible to choose the element formulations used in ADINA version 8.7, using the command `KINEMATICS BEAM-ALGORITHM=V87`. (However, if the elasto-plastic material model is chosen, then the current beam element kinematics and material algorithms are always used.)

The results obtained will vary depending upon whether the current formulations or the 8.7 formulations are used.

Table 2.4-1: Table of beam capabilities

Formulation and material model	Cross-section PROPERTIES	Cross-sections RECTANGULAR, PIPE	Cross-section BOX	Cross-sections U, I, L
Linear elastic	✓ ¹	✓ ²	✓ ³	✓ ⁴
Large displacement elastic	✓ ¹	✓ ²	✓ ³	✓ ⁴
MNO plastic	---	✓ ²	✓ ³	✓ ⁴
Large displacement plastic	---	✓ ²	✓ ³	✓ ⁴
MNO moment-curvature	✓ ²	---	---	---
Large displacement moment-curvature	✓ ²	---	---	---

1. Either the standard or warping beam can be used. However, if the properties correspond to an open thin-walled section, the warping beam is recommended for general use.
2. Only the standard beam can be used.
3. Either the standard or warping beam can be used.
4. Either the standard or warping beam can be used, however the warping beam element is recommended for general use.

2.4.1 Beam geometry

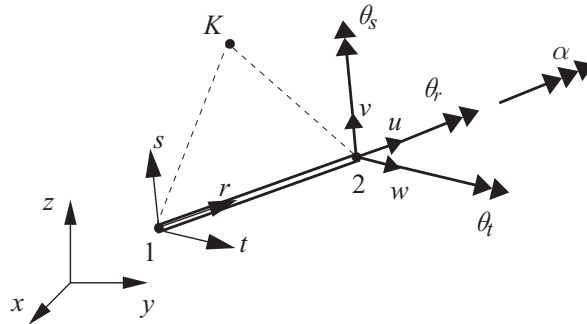
- Fig 2.4-1 shows the beam element along with its local coordinate system (r,s,t) . The r direction always lies along the neutral line of the beam. The orientations of the s and t directions are defined using either an auxiliary node K (as in Fig. 2.4-1(a)) or an orientation vector (XO,YO,ZO) (as in Fig. 2.4-1(b)). If node K is specified in the input, it is always used as the auxiliary node. If node K is not specified, then the orientation vector is used.

When the orientation vector is used, the components (XO,YO,ZO) are interpreted either in the skew system of local node 1 (the default), or in the global system.

Notice that, for the Hermitian beam element, (r,s,t) are not isoparametric coordinates, rather (r,s,t) have the same units as the global coordinates.

Fig 2.4-1 also shows the degrees of freedom at the local nodes. These degrees of freedom are defined in the local coordinate system. The α degree of freedom is used only for the warping beam element.

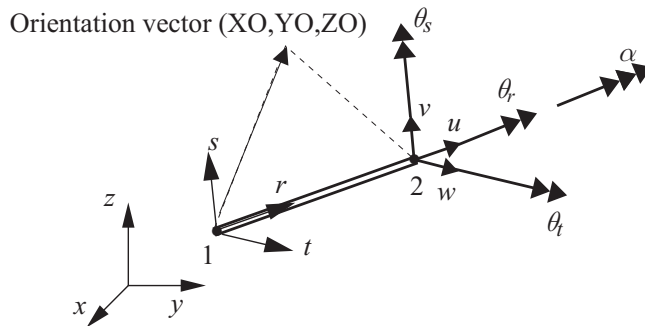
- The forces/moments in the beam element are shown in Fig. 2.4-2. These forces/moments are also defined in the element local coordinate system. The bimoment forces are used only for the warping beam element.
- The s and t directions give the orientation of the beam element cross-section. Care must be used in defining the s and t directions so that the beam element cross-section has the desired orientation.



The s -direction lies in the plane defined by nodes 1, 2, K , and the t -direction is perpendicular to the r - s plane.

α : warping DOF

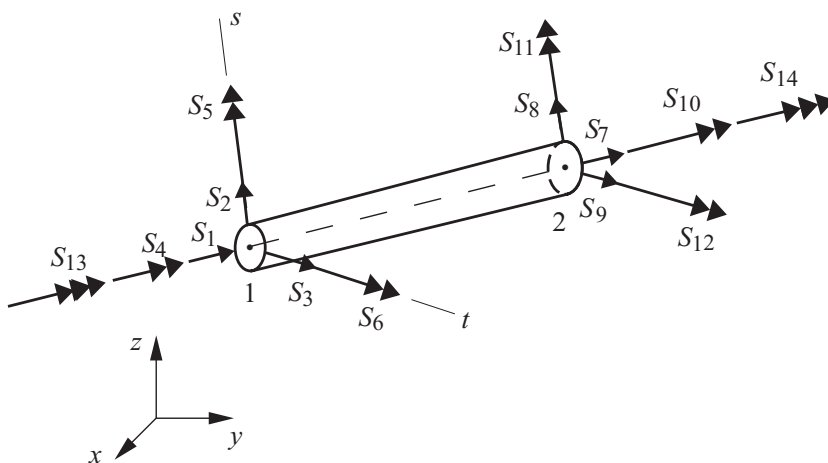
(a) Geometry definition, using auxiliary node K



The s -direction lies in the plane defined by nodes 1, 2 and the orientation vector. The t -direction is perpendicular to the r - s plane.

(b) Geometry definition, using orientation vector

Figure 2.4-1: Degrees of freedom and local axes for beam element



S_1 = r -direction force at node 1 (axial force, positive in compression)

S_2 = s -direction force at node 1 (shear force)

S_3 = t -direction force at node 1 (shear force)

S_4 = r -direction moment at node 1 (torsion)

S_5 = s -direction moment at node 1 (bending moment)

S_6 = t -direction moment at node 1 (bending moment)

S_7 = r -direction force at node 2 (axial force, positive in tension)

S_8 = s -direction force at node 2 (shear force)

S_9 = t -direction force at node 2 (shear force)

S_{10} = r -direction moment at node 2 (torsion)

S_{11} = s -direction moment at node 2 (bending moment)

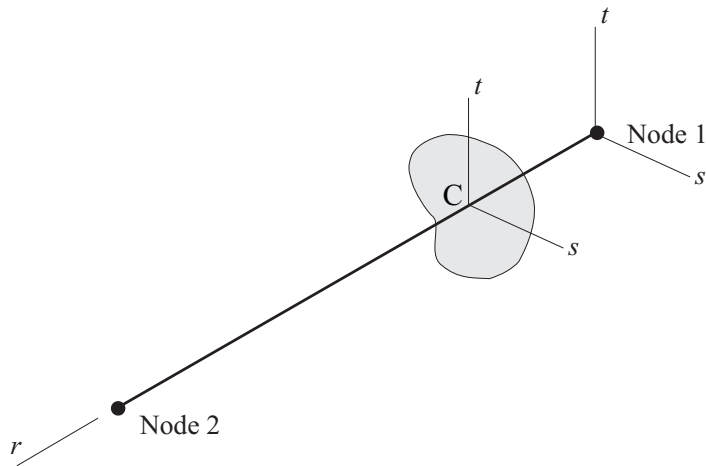
S_{12} = t -direction moment at node 2 (bending moment)

S_{13} = bimoment at node 1

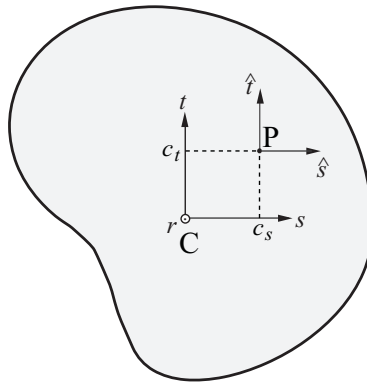
S_{14} = bimoment at node 2

Figure 2.4-2: Element end forces/moments

- The beam element cross-section with its coordinate system is shown in Fig. 2.4-3. The cross-section coordinate system has its origin on the line connecting the beam end-nodes. (Note that the cross-section coordinate system uses the same letters (s, t) as the beam element coordinate system, however this should not cause confusion.)



(a) Isometric view



C: Centroid of the cross-section

P: Shear center of the cross-section

(b) s - t plane view, showing location of shear center

Figure 2.4-3: Beam cross-section coordinate system

Fig 2.4-3(b) shows the cross-section with its centroid and shear center. The centroid of the cross-section is assumed to be at the origin of the (s, t) system. Also notice that a coordinate system

(\hat{s}, \hat{t}) is introduced, with origin at the shear center. The (\hat{s}, \hat{t}) system is always parallel to the (s, t) system, and the origin of the (\hat{s}, \hat{t}) is located at $(s, t) = (c_s, c_t)$.

- The shear center is also the center of twist.

2.4.2 Beam cross-section geometric properties

The following cross-sectional geometric properties are listed here for reference.

c_s, c_t : s and t coordinates of the shear center with respect to the centroid (terms CSOFFSET, CTOFFSET)

A_s^{sh}, A_t^{sh} : effective shear cross-section areas in the s and t directions (terms SAREA, TAREA) . Note, these terms are actual areas, with units of length squared.

$\phi = \phi(s, t)$: warping function, see description below.

Governing equations for location of beam centroid:

$$\int s dA = \int t dA = 0 \quad (2.4-1)$$

Normalization equation for warping function:

$$\int \phi dA = 0 \quad (2.4-2)$$

Governing equations for location of beam shear center:

$$\int s \phi dA = \int t \phi dA = 0 \quad (2.4-3)$$

A : Cross-sectional area (term AREA)

$$A = \int dA \quad (2.4-4)$$

I_{ss} : Inertia for bending about the s -axis (term SINERTIA):

$$I_{ss} = \int t^2 dA \quad (2.4-5)$$

I_{tt} : Inertia for bending about the t-axis (term TINERTIA):

$$I_{tt} = \int s^2 dA \quad (2.4-6)$$

I_{st} : Product of inertia (term STINERTIA):

$$I_{st} = \int st dA \quad (2.4-7)$$

J : Saint-Venant torsional constant (term RINERTIA):

$$J = \int \left((\phi_{,s} - \hat{t})^2 + (\phi_{,t} + \hat{s})^2 \right) dA \quad (2.4-8)$$

(do not confuse the Saint-Venant torsional constant with the polar moment of inertia)

I_{ω} : Warping constant (term WINERTIA):

$$I_{\omega} = \int \phi^2 dA \quad (2.4-9)$$

(this term is also referred to as C_{ω} in handbooks)

A_{15} : Wagner effect constant (term RRINERTIA)

$$A_{15} = \int (\hat{s}^2 + \hat{t}^2) dA \quad (2.4-10)$$

A_{25} : Wagner effect constant (term SRINERTIA)

$$A_{25} = \int s (\hat{s}^2 + \hat{t}^2) dA \quad (2.4-11)$$

A_{35} : Wagner effect constant (term TRINERTIA)

$$A_{35} = \int t (\hat{s}^2 + \hat{t}^2) dA \quad (2.4-12)$$

A_{45} : Wagner effect constant (term WRINERTIA)

$$A_{45} = \int \phi (\hat{s}^2 + \hat{t}^2) dA \quad (2.4-13)$$

A_{55} : Wagner effect constant (term DRINERTIA)

$$A_{55} = \int (\hat{s}^2 + \hat{t}^2)^2 dA \quad (2.4-14)$$

The term names are the names used in the CROSS-SECTION PROPERTIES command. Also note that the terminology

$$()_{,s} = \frac{\partial()}{\partial s}, ()_{,t} = \frac{\partial()}{\partial t} \text{ is used.}$$

2.4.3 Beam formulation

- The formulation of the beam is a generalization of the Euler-Bernoulli beam formulation.
- The cross-section of the beam is assumed to be rigid in its own plane so no distortion of the cross-section is considered. However, the cross-section can warp out of its plane.
- The two nodes and the origin of the local coordinate system (s, t) are located at the centroid of the beam cross-section. The local coordinate system of the element may or may not be coincident with the principal planes of inertia of the cross-section.

A brief summary of the beam formulation derivation is now given.

2.4.3.1 Beam displacements

- The beam displacements in the beam local coordinate system are u (axial r -direction displacement), v (transverse s -direction displacement) and w (transverse t -direction displacement).
- The displacements at an arbitrary point on the beam cross-section can be written in terms of the displacements, rotations and warping measured at the beam neutral axis:

$$\begin{aligned} u &= u_o - s\theta_t + t\theta_s + \phi\alpha \\ v &= v_n - t\theta_r \\ w &= w_n + s\theta_r \end{aligned} \quad (2.4-15)$$

In this expression, u_o is the average longitudinal displacement of

the cross-section, v_n , w_n are the displacements of the beam neutral axis (corresponding to point C in Fig. 2.4-3), θ_r , θ_s , θ_t are the rotations of the section about the r , s , t axes, $\phi = \phi(s, t)$ is the warping function, and α is the warping function multiplier. The warping function and warping function multiplier will be discussed in greater detail below.

Equation (2.4-15) is seen to include the assumptions that the cross-section translates rigidly, rotates rigidly and warps (due to the warping function). In all cases, the translations and rotations, with respect to the beam local coordinate system, are assumed to be small. (Later on, we will allow for large displacements by allowing the local coordinate system to translate and rotate.)

- Under the assumption of unit warping, the axial displacement due to torsion is given by

$$u = \phi(s, t)\theta_{r,r} \quad (2.4-16)$$

where $\phi_{,r} = \frac{\partial \phi}{\partial r}$. Hence $\alpha = \theta_{r,r}$ and (2.4-15) can be written

$$\begin{aligned} u &= u_o - s\theta_t + t\theta_s + \phi\theta_{r,r} \\ v &= v_n - t\theta_r \\ w &= w_n + s\theta_r \end{aligned} \quad (2.4-17)$$

The dimension of α is seen to be $[L^{-1}]$.

- It will be convenient to express the transverse displacements in terms of the transverse displacements at the shear center. The transverse displacements at the shear center (corresponding to point P in Fig. 2.4-3) are

$$\begin{aligned} v_p &= v_n - c_t\theta_r \\ w_p &= w_n + c_s\theta_r \end{aligned} \quad (2.4-18)$$

Therefore

$$\begin{aligned}v &= v_p - \hat{t}\theta_r \\w &= w_p + \hat{s}\theta_r\end{aligned}\tag{2.4-19}$$

- We now introduce the assumption that the cross-section remains normal to the shear center axis:

$$\begin{aligned}\theta_s &= -w_{p,r} \\ \theta_t &= v_{p,r}\end{aligned}\tag{2.4-20}$$

This assumption corresponds to the use of Euler-Bernoulli beam theory, however here the shear center axis is used instead of the more commonly-used neutral axis. This choice will be explained below.

2.4.3.2 Displacement derivatives and strains

The derivatives of the displacements with respect to the r, s, t directions are

$$\begin{aligned}u_{,r} &= u_{o,r} - s\theta_{t,r} + t\theta_{s,r} + \phi\theta_{r,rr} \\ u_{,s} &= -\theta_t + \phi_{,s}\theta_{r,r} \\ u_{,t} &= \theta_s + \phi_{,t}\theta_{r,r}\end{aligned}\tag{2.4-21}$$

$$\begin{aligned}v_{,r} &= \theta_t - \hat{t}\theta_{r,r} \\ v_{,s} &= 0 \\ v_{,t} &= -\theta_r\end{aligned}\tag{2.4-22}$$

$$\begin{aligned}w_{,r} &= -\theta_s + \hat{s}\theta_{r,r} \\ w_{,s} &= \theta_r \\ w_{,t} &= 0\end{aligned}\tag{2.4-23}$$

in which (2.4-20) has been used.

Considering only engineering strains for now,

$$\begin{aligned}
e_{rr} &= u_{,r} = u_{o,r} - s\theta_{t,r} + t\theta_{s,r} + \phi\theta_{r,rr} \\
\gamma_{rs} &= u_{,s} + v_{,r} = (\phi_{,s} - \hat{t})\theta_{r,r} \\
\gamma_{rt} &= u_{,t} + w_{,r} = (\phi_{,t} + \hat{s})\theta_{r,r}
\end{aligned} \tag{2.4-24}$$

Equations (2.4-24) show that shear strains arise only due to torsion. This is the reason that we have used the shear center instead of the neutral axis in the above developments.

2.4.3.3 Stresses

It is assumed that the only nonzero stresses are $\tau_{rr}, \tau_{rs}, \tau_{rt}$. The stresses are computed from the strains using the material law, either elastic or elasto-plastic. For a linear elastic material

$$\tau_{rr} = Ee_{rr}, \tau_{rs} = G\gamma_{rs}, \tau_{rt} = G\gamma_{rt} \tag{2.4-25}$$

where E, G are the Young's and shear moduli, respectively.

2.4.3.4 Principle of virtual work

The virtual work can be written

$$\delta W = \int (\tau_{rr}\delta e_{rr} + \tau_{rs}\delta\gamma_{rs} + \tau_{rt}\delta\gamma_{rt}) dV \tag{2.4-26}$$

The force vector and stiffness matrix can be derived from this expression, once a beam interpolation has been chosen. In general, numerical integration must be used, however when the material is linear elastic, the integrals are evaluated analytically.

2.4.3.5 Torsional response

In order to gain additional insight into the torsional response, we consider a beam loaded by an external axial torque. The displacements of the beam are

$$\begin{aligned}
u &= \phi \theta_{r,r} \\
v &= \hat{t} \theta_r \\
w &= -\hat{s} \theta_r
\end{aligned} \tag{2.4-27}$$

the corresponding engineering strains are

$$\begin{aligned}
e_{rr} &= \phi \theta_{r,rr} \\
\gamma_{rs} &= (\phi_{,s} - \hat{t}) \theta_{r,r} \\
\gamma_{rt} &= (\phi_{,t} + \hat{s}) \theta_{r,r}
\end{aligned} \tag{2.4-28}$$

and the principle of virtual work can be written

$$\int (\tau_{rr} \delta e_{rr} + \tau_{rs} \delta \gamma_{rs} + \tau_{rt} \delta \gamma_{rt}) dV = T \delta \theta_r \Big|_0^L \tag{2.4-29}$$

where T is the externally applied torque at the ends of the beam.
The left-hand-side terms can be written as

$$\int \tau_{rr} \delta e_{rr} dV = \int \tau_{rr} \phi \theta_{r,rr} dV \tag{2.4-30}$$

$$\int (\tau_{rs} \delta \gamma_{rs} + \tau_{rt} \delta \gamma_{rt}) dV = \int \left[\tau_{rs} (\phi_{,s} - \hat{t}) + \tau_{rt} (\phi_{,t} + \hat{s}) \right] \delta \theta_{r,r} dV \tag{2.4-31}$$

Define

$$\begin{aligned}
M_b &= \int \tau_{rr} \phi dA, \quad F_b = \int (\tau_{rs} \phi_{,s} + \tau_{rt} \phi_{,t}) dA, \\
M_r &= \int (-\tau_{rs} \hat{t} + \tau_{rt} \hat{s}) dA
\end{aligned} \tag{2.4-32a,b,c}$$

where M_b is the bimoment (some authors insert a minus sign), F_b is the bishear and M_r is the torque calculated from the shear stresses. Then the principle of virtual work can be written

$$\int (M_b \delta \theta_{r,rr} + (F_b + M_r) \delta \theta_{r,r}) dr = T \delta \theta_r \Big|_0^L \quad (2.4-33)$$

Using the calculus of variations, the principle of virtual work can be rewritten as

$$-\int [F_b + M_r - M_{b,r}]_r \delta \theta_r dr = \left[T - (F_b + M_r - M_{b,r}) \right] \delta \theta_r \Big|_0^L - M_b \delta \theta_{r,r} \Big|_0^L \quad (2.4-34)$$

and we see the following:

- 1) The bimoment is energy-conjugate to the warping multiplier (recall $\alpha = \theta_{r,r}$)
- 2) If the warping multiplier is free at an end, the bimoment is zero at that end.
- 3) The torque is equilibrated by the quantity $F_b + M_r - M_{b,r}$.
- 4) If the warping multiplier is free at both ends, the bimoment is zero throughout and the external torque is equilibrated by the sum of the bishear and the torque calculated from the shear stresses.

2.4.3.6 Strain energy for a linear elastic material

We now return to Section 2.4.3.4 and consider a linear elastic material. The strain energy in the beam can be written

$$W = \frac{1}{2} E \int e_{rr}^2 dV + \frac{1}{2} G \left(\int \gamma_{rs}^2 + \gamma_{rt}^2 \right) dV \quad (2.4-35)$$

Substituting from (2.4-24) gives

$$\begin{aligned} W = & \frac{1}{2} E \int (u_{o,r} - s \theta_{t,r} + t \theta_{s,r} + \phi \theta_{r,rr})^2 dV \\ & + \frac{1}{2} G \int \left((\phi_{,s} - \hat{t})^2 + (\phi_{,t} + \hat{s})^2 \right) \theta_{r,r}^2 dV \end{aligned} \quad (2.4-36)$$

This can be expanded to give

$$\begin{aligned}
W = & \frac{1}{2} E \left\{ \int u_{o,r}^2 dV - 2 \int u_{o,r} \theta_{t,r} s dV + 2 \int u_{o,r} \theta_{s,r} t dV + 2 \int u_{o,r} \theta_{r,rr} \phi dV \right. \\
& + \int \theta_{t,r}^2 s^2 dV - 2 \int \theta_{t,r} \theta_{s,r} s t dV - 2 \int \theta_{t,r} \theta_{r,rr} s \phi dV \\
& + \int \theta_{s,r}^2 t^2 dV + 2 \int \theta_{s,r} \theta_{r,rr} t \phi dV \\
& \left. + \int \theta_{r,rr}^2 \phi^2 dV \right\} \\
& + \frac{1}{2} G \int \theta_{r,r}^2 \left((\phi_{,s} - \hat{t})^2 + (\phi_{,t} + \hat{s})^2 \right) dV
\end{aligned} \tag{2.4-37}$$

We now recognize that $\int dV = \int \int dA dr$, where $\int dA$ represents an integral over the beam cross-section. Also since all terms such as $u_{o,r}$, $\theta_{s,r}$, etc. are constant over the cross-section, (2.4-37) can be written as

$$\begin{aligned}
W = & \frac{1}{2} E \left\{ \int dA \int u_{o,r}^2 dr - 2 \int s dA \int u_{o,r} \theta_{t,r} dr + 2 \int t dA \int u_{o,r} \theta_{s,r} dr + 2 \int \phi dA \int u_{o,r} \theta_{r,rr} dr \right. \\
& + \int s^2 dA \int \theta_{t,r}^2 dr - 2 \int s t dA \int \theta_{t,r} \theta_{s,r} dr - 2 \int s \phi dA \int \theta_{t,r} \theta_{r,rr} dr \\
& + \int t^2 dA \int \theta_{s,r}^2 dr + 2 \int t \phi dA \int \theta_{s,r} \theta_{r,rr} dr \\
& \left. + \int \phi^2 dA \int \theta_{r,rr}^2 dr \right\} \\
& + \frac{1}{2} G \int \left((\phi_{,s} - \hat{t})^2 + (\phi_{,t} + \hat{s})^2 \right) dA \int \theta_{r,r}^2 dr
\end{aligned} \tag{2.4-38}$$

By definition of the neutral axis, $\int s dA = \int t dA = 0$, and by definition of the warping function, $\int \phi dA = \int s \phi dA = \int t \phi dA = 0$. Therefore, in terms of the geometric properties given in Section 2.4.2, the strain energy of the beam can be written

$$\begin{aligned}
W = & \frac{1}{2} E \left\{ A \int u_{o,r}^2 dr + I_{tt} \int \theta_{t,r}^2 dr - 2 I_{st} \int \theta_{t,r} \theta_{s,r} dr + I_{ss} \int \theta_{s,r}^2 dr + I_{\omega} \int \theta_{r,rr}^2 dr \right\} \\
& + \frac{1}{2} GJ \int \theta_{r,r}^2 dr
\end{aligned}$$

(2.4-39)

This expression can be integrated analytically once beam interpolations are introduced (see Section 2.4.4). And the force vector and stiffness matrix are obtained by taking variations of the strain energy.

2.4.3.7 Wagner effect

H. Wagner discussed the mechanics of thin-walled open sections in the following reference:

ref. Wagner, H., *Verdrehung und Knickung von offenen Profilen* (Torsion and Buckling of Open Sections), 25th Anniversary Publication, Technische Hochschule, Danzig, 1904-1929, translated as N.A.C.A. Technical Memorandum No. 807, National Committee for Aeronautics, 1936.

The following observation in this paper is of significance: Consider a beam carrying axial stress, and consider a small twist of the beam. The axial stress now has a transverse component due to the twist. The resultant of all of the transverse components of axial stress acting on a cross-section has a non-zero twisting moment.

For brevity, we term the above observation, the "Wagner effect".

Fig. 2.4-4 shows a schematic. The beam carries an initial axial stress (this stress can vary from point to point in the cross-section). For the material fiber shown in the figure, this stress has magnitude $\tau_{rr} = T$ and is oriented in direction $(1, 0, 0)$. Now suppose that the cross-section undergoes a small twist, of magnitude α per unit length. After deformation, the material fiber has the orientation $(1, -\hat{t}\alpha, \hat{s}\alpha)$ and the Cauchy stress tensor corresponding to T is

$$\boldsymbol{\tau} = \begin{bmatrix} 1 & -\hat{t}\alpha & \hat{s}\alpha \\ -\hat{t}\alpha & 0 & 0 \\ \hat{s}\alpha & 0 & 0 \end{bmatrix} T \quad (2.4-40)$$

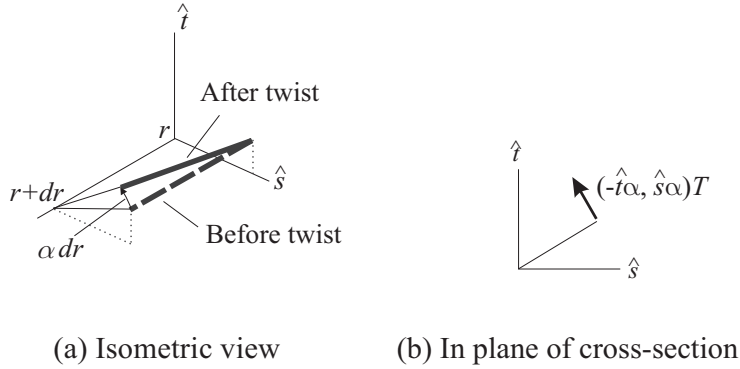


Figure 2.4-4: Wagner effect

The twisting moment in the plane of the cross-section due to the stress at the point (\hat{s}, \hat{t}) is $M_r = (\hat{s}^2 + \hat{t}^2)T\alpha$, and the virtual work done by a virtual twist $\delta\alpha$ is

$$M_r \delta\alpha = (\hat{s}^2 + \hat{t}^2)T\alpha \delta\alpha \quad (2.4-41)$$

This result can also be obtained by using the proper continuum mechanics stress and strain measures. Following standard continuum mechanics procedures, the Green-Lagrange strain component in the axial direction, due to the twist α , is

$$\varepsilon_{rr} = \frac{1}{2}(\hat{s}^2 + \hat{t}^2)\alpha^2 \quad (2.4-42)$$

and the 2nd Piola-Kirchhoff stress tensor is

$$\mathbf{S} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} T \quad (2.4-43)$$

Thus the work done by the axial stress as a result of the twist is simply

$$S_{rr} \delta\varepsilon_{rr} = (\hat{s}^2 + \hat{t}^2)T\alpha \delta\alpha \quad (2.4-44)$$

Since (2.4-41) and (2.4-44) are equal, the Wagner effect can be included simply by using the Green-Lagrange strain tensor instead of the small strain tensor, and interpreting the stresses as 2nd Piola-Kirchhoff stresses instead of Cauchy stresses.

- It is clear that the Wagner effect is a geometrically nonlinear effect. The Wagner effect is included only in conjunction with the large displacement formulation described in Section 2.4.5.3.
- The procedure of using the Green-Lagrange strain tensor instead of the small strain tensor, and interpreting the stresses as 2nd Piola-Kirchhoff stresses instead of Cauchy stresses, works extremely well when used in the large displacement/small strain formulations for continuum finite elements. However, it should be remembered that the geometry of continuum elements is completely specified by the nodes, whereas the geometry of beam elements is specified in terms of cross-sectional properties such as A , I_{ss} , etc. Including all terms in the Green-Lagrange strain tensor leads to a very large number of cross-sectional properties.
- Therefore we focus on including only the minimum number of additional terms in the strain tensor so that the Wagner effect is properly included. Recalling that the components of the Green-Lagrange strain tensor are

$$\begin{aligned}\varepsilon_{rr} &= e_{rr} + \frac{1}{2}(u_{,r}^2 + v_{,r}^2 + w_{,r}^2) \\ 2\varepsilon_{rs} &= \gamma_{rs} + (u_{,r}u_{,s} + v_{,r}v_{,s} + w_{,r}w_{,s}) \\ 2\varepsilon_{rt} &= \gamma_{rt} + (u_{,r}u_{,t} + v_{,r}v_{,t} + w_{,r}w_{,t})\end{aligned}\tag{2.4-45}$$

we will assume that all products in ε_{rs} , ε_{rt} can be neglected compared to γ_{rs} , γ_{rt} , and that $u_{,r}^2$ can be neglected compared to e_{rr} .

- In order to neglect additional terms, we need to use the fact that the Wagner effect is implemented along with the large displacement formulation described in Section 2.4.5.3. In the large

displacement formulation, all angles are measured with respect to a coordinate system attached to the beam. In the limit of mesh refinement, these angles can be made arbitrarily small (again, measured with respect to the coordinate system attached to the beam).

Now consider the term $v_{,r}^2 = \theta_t^2 - 2\hat{t}\theta_t\theta_{r,r} + \hat{t}^2\theta_{r,r}^2$. In the limit of mesh refinement, θ_t becomes arbitrarily small, however $\theta_{r,r}$ cannot be made arbitrarily small by mesh refinement. A similar statement can be used for term $w_{,r}^2$.

So we use the approximations

$$\begin{aligned} v_{,r}^2 &\approx \hat{t}^2\theta_{r,r}^2 \\ w_{,r}^2 &\approx \hat{s}^2\theta_{r,r}^2 \end{aligned} \tag{2.4-46}$$

- The result is that the components of the Green-Lagrange strain tensor are approximated by

$$\begin{aligned} \varepsilon_{rr} &\approx e_{rr} + \frac{1}{2}(\hat{s}^2 + \hat{t}^2)\theta_{r,r}^2 = u_{o,r} - s\theta_{t,r} + t\theta_{s,r} + \phi\theta_{r,rr} + \frac{1}{2}(\hat{s}^2 + \hat{t}^2)\theta_{r,r}^2 \\ 2\varepsilon_{rs} &\approx \gamma_{rs} = (\phi_{,s} - \hat{t})\theta_{r,r} \\ 2\varepsilon_{rt} &\approx \gamma_{rt} = (\phi_{,t} + \hat{s})\theta_{r,r} \end{aligned} \tag{2.4-47}$$

and the virtual work becomes

$$\delta W = \int S_{rr}\delta\varepsilon_{rr} + S_{rs}\delta\gamma_{rs} + S_{rt}\delta\gamma_{rt} dV \tag{2.4-48}$$

- Constructing the strain energy (for a linear elastic material) using the procedure given above leads to additional terms in the strain energy:

$$\begin{aligned}
W = \dots + \frac{1}{2} E \Big\{ & \int (\hat{s}^2 + \hat{t}^2) dA \int u_{o,r} \theta_{r,r}^2 dr \\
& - \int s (\hat{s}^2 + \hat{t}^2) dA \int \theta_{t,r} \theta_{r,r}^2 dr \\
& + \int t (\hat{s}^2 + \hat{t}^2) dA \int \theta_{s,r} \theta_{r,r}^2 dr \\
& + \int \phi (\hat{s}^2 + \hat{t}^2) dA \int \theta_{r,rr} \theta_{r,r}^2 dr \\
& + \frac{1}{4} \int (\hat{s}^2 + \hat{t}^2)^2 dA \int \theta_{r,r}^4 dr \Big\}
\end{aligned} \tag{2.4-49}$$

where the ... are the terms from the small strain theory (2.4-38).

Using the geometric properties given in Section 2.4.2, the strain energy for a linear elastic material can be written

$$\begin{aligned}
W = \frac{1}{2} E \Big\{ & A \int u_{o,r}^2 dr + I_u \int \theta_{t,r}^2 dr - 2I_{st} \int \theta_{t,r} \theta_{s,r} dr + I_{ss} \int \theta_{s,r}^2 dr + I_{\omega} \int \theta_{r,rr}^2 dr \Big\} \\
& + \frac{1}{2} E \Big\{ A_{15} \int u_{o,r} \theta_{r,r}^2 dr - A_{25} \int \theta_{t,r} \theta_{r,r}^2 dr + A_{35} \int \theta_{s,r} \theta_{r,r}^2 dr + A_{45} \int \theta_{r,rr} \theta_{r,r}^2 dr \\
& \quad + \frac{1}{4} A_{55} \int \theta_{r,r}^4 dr \Big\} \\
& + \frac{1}{2} GJ \int \theta_{r,r}^2 dr
\end{aligned} \tag{2.4-50}$$

This expression can be integrated analytically once beam interpolations are introduced (see Section 2.4.4). And the force vector and stiffness matrix are obtained by taking variations of the strain energy.

- The Wagner effect is implemented for inelastic materials simply by computing the strains using (2.4-47), and by including the proper geometrically nonlinear terms in the stiffness matrix.
- As a very simple example in which the Wagner effect is important, consider the axial loading of an elastic beam, in which bending is suppressed but in which torsion is allowed. For simplicity the angle of twist per unit length $\theta_{r,r}$ is assumed to be constant. (2.4-50) becomes

$$W = \frac{1}{2} EA \int u_{o,r}^2 dr + \frac{1}{2} EA_{15} \int u_{o,r} \theta_{r,r}^2 dr + \frac{1}{2} GJ \int \theta_{r,r}^2 dr + \frac{1}{8} EA_{55} \int \theta_{r,r}^4 dr \quad (2.4-51)$$

Taking the variation of W with respect to $\theta_{r,r}$ and using

$$\tau_{rr} \approx S_{rr} = Eu_{o,r} \text{ gives}$$

$$\delta W = \left(\int (A_{15} \tau_{rr} + GJ) \theta_{r,r} dr + \frac{1}{2} EA_{55} \int \theta_{r,r}^3 dr \right) \delta \theta_{r,r} \quad (2.4-52)$$

For small twist, the $\theta_{r,r}^3$ term can be neglected. Equation (2.4-52)

shows that for stress $\tau_{rr} \leq -G \frac{J}{A_{15}}$, the equation of motion

becomes unstable, that is, torsional buckling can occur. The ratio

$\frac{J}{A_{15}}$ influences how important the Wagner effect is for a given

cross-section. For a pipe section, both J and A_{15} are equal to the polar moment of inertia of the section, so torsional buckling can only occur for compressive stresses that are much larger than can reasonably be expected. On the other hand, for a thin-walled open

section, the ratio $\frac{J}{A_{15}}$ can be much smaller than unity, then

torsional buckling can occur for much smaller compressive stresses.

Of course, the simplifications that we made in obtaining (2.4-

52) mean that $\tau_{rr} = -G \frac{J}{A_{15}}$ is not the torsional buckling load in

general. For actual formulas involving torsional and flexural buckling loads, see references such as

ref. A. Gjelsvik, *The Theory of Thin Walled Bars*, John Wiley & Sons, 1981

2.4.3.8 Kinetic energy and the mass matrix

The kinetic energy of the beam can be written

$$K = \frac{1}{2} \rho \int (\dot{u}^2 + \dot{v}^2 + \dot{w}^2) dV \quad (2.4-53)$$

Substituting from (2.4-17) gives

$$K = \frac{1}{2} \rho \left\{ A \int (\dot{u}_o^2 + \dot{v}_n^2 + \dot{w}_n^2) dr + I_u \int (\dot{\theta}_r^2 + \dot{\theta}_t^2) dr - 2I_{st} \int \dot{\theta}_t \dot{\theta}_s dr \right. \\ \left. + I_{ss} \int (\dot{\theta}_r^2 + \dot{\theta}_s^2) dr + I_\omega \int \dot{\theta}_{r,r}^2 dr \right\} \quad (2.4-54)$$

The mass matrix is obtained from (2.4-54), once a beam interpolation has been introduced.

- The rotary inertias and the warping constant can have a significant effect on the torsional response of the beam, particularly for thin-walled open sections.

2.4.3.9 The warping function

- The torsional response of beams significantly depends on the type of their cross-section (e.g., solid section, thin-walled open section, thin-walled closed section).
- The torsional response is closely related to the displacements of the beam cross-section that are out of the plane of the cross-section (see Fig. 2.4-5). These displacements are termed "warping displacements", since torsion causes warping of the beam (see Fig. 2.4-5).
- The warping displacements are included in the beam formulation through the use of the warping function $\phi(s, t)$. The warping function must be determined in advance for each cross-section.
- For circular cross-sections, torsion does not cause warping displacements, thus the warping function is zero for circular cross-sections.
- For the solid rectangular section undergoing plasticity, warping is accounted for as discussed in Section 2.4.6.2. Note that it is not

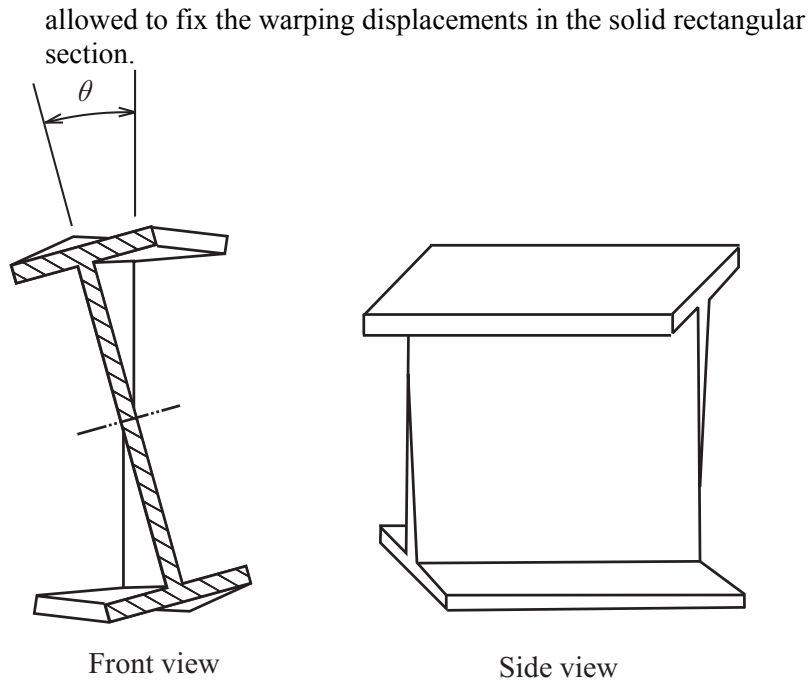


Figure 2.4-5: Torsional warping of an I-beam subjected to pure torsion

- In the following discussion, we focus on warping functions for thin-walled sections. The theory of the warping function for thin-walled sections can be found in several references, including

ref. A. Gjelsvik, *The Theory of Thin Walled Bars*, John Wiley & Sons, 1981

A brief summary of the theory follows, in which we have adjusted the notations so as not to conflict with the notations used in the rest of this section.

Open thin-walled sections

Consider a generic open thin-walled section (Fig 2.4-6). The centerline of the section is called the contour. The coordinate through the section thickness is denoted p and the coordinate along the contour is denoted q .

We introduce an additional point, not necessarily in the cross-section, called a pole (Fig 2.4-7). Corresponding to a point in the cross-section and the pole P, there is a coordinate system \hat{p}, \hat{q} , in which \hat{p} is parallel to dp and \hat{q} is parallel to dq .

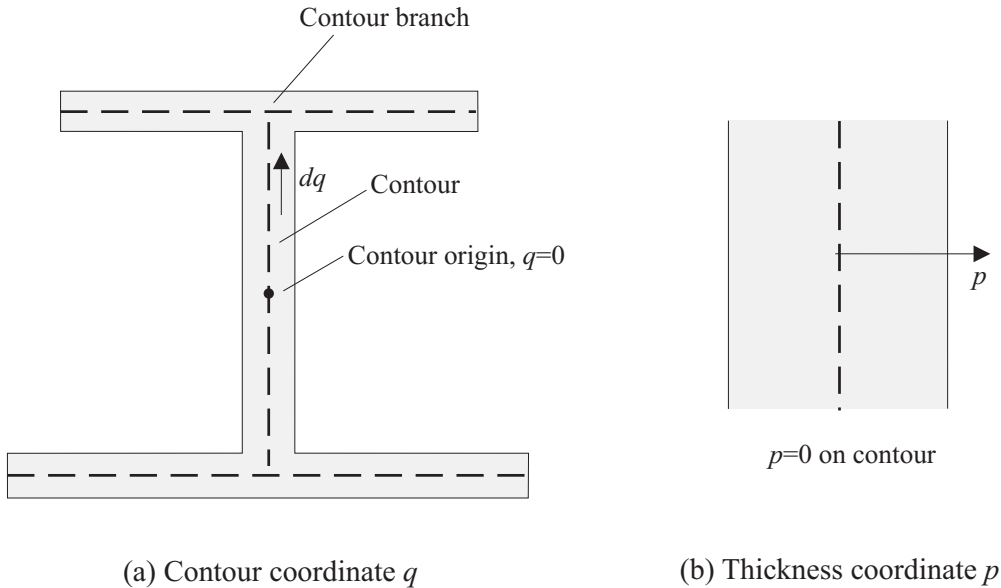


Figure 2.4-6: Cross-section coordinate system

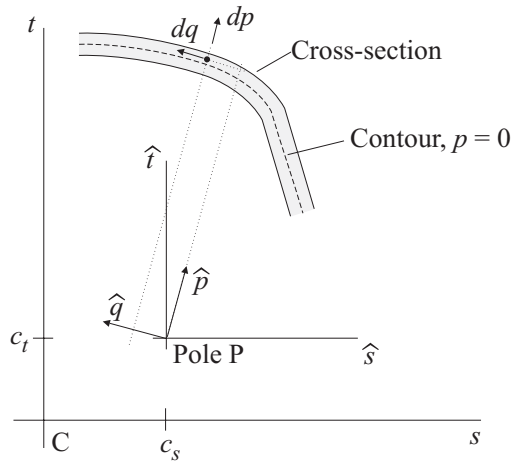


Figure 2.4-7: Notation used in construction of warping function

Now consider a rigid body rotation of the cross-section about pole P (Fig 2.4-7). Clearly

$$\begin{aligned} u_{\hat{s}} &= -\hat{t}\theta_r, \\ u_{\hat{t}} &= \hat{s}\theta_r \end{aligned} \quad (2.4-55)$$

where $u_{\hat{s}}, u_{\hat{t}}$ are the displacements in the \hat{s}, \hat{t} directions. Also

$$\begin{aligned} u_{\hat{p}} &= -\hat{q}\theta_r, \\ u_{\hat{q}} &= \hat{p}\theta_r \end{aligned} \quad (2.4-56)$$

where $u_{\hat{p}}, u_{\hat{q}}$ are the displacements in the \hat{p}, \hat{q} directions. Since \hat{p}, \hat{q} are parallel to dp, dq ,

$$\begin{aligned} u_p &= -\hat{q}\theta_r, \\ u_q &= \hat{p}\theta_r \end{aligned} \quad (2.4-57)$$

are the displacements perpendicular to and parallel to the contour.

From this information, we can construct an approximate warping function $\phi(s, t)$. First consider the shear strain

$$\gamma_{rq} = u_{r,q} + u_{q,r} \quad (2.4-58)$$

where u_r is the displacement out of the plane of the cross-section (the warping displacement). The choice

$$u_{r,q} = -u_{q,r} = -\hat{p} \theta_{r,r} \quad (2.4-59)$$

sets $\gamma_{rq} = 0$ on the contour. Now consider the shear strain

$$\gamma_{rp} = u_{r,p} + u_{p,r} \quad (2.4-60)$$

The choice

$$u_{r,p} = -u_{p,r} = \hat{q} \theta_{r,r} \quad (2.4-61)$$

sets $\gamma_{rp} = 0$ along lines perpendicular to the contour. Since the cross-section is thin, \hat{q} is constant along these lines, so

$$u_r|_{(p,q)} = u_r|_{(0,q)} + p \hat{q} \theta_{r,r} \quad (2.4-62)$$

and therefore

$$u_r|_{(p,q)} = \left(-\int_0^q \hat{p} dq + p \hat{q} \right) \theta_{r,r} \quad (2.4-63)$$

From this expression, we observe that the warping function is

$$\phi(s, t) = -\int_0^q \hat{p} dq + p \hat{q} \quad (2.4-64)$$

Thus the total warping is composed of two contributions, the first contribution along the centerline, called contour warping, and the second contribution through the thickness, called thickness warping.

From the above conditions on the shear strain, the shear strain caused by torsion is aligned with the q direction and proportional to

p .

- If the cross-section of the beam only consists of thin rectangular plates all intersecting at a common point, then the contour warping is zero and the warping function only contains thickness warping (see Fig. 2.4-8).

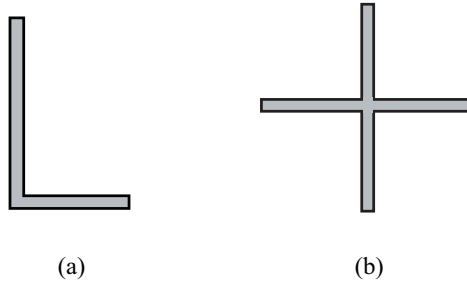


Figure 2.4-8: Sections with negligible contour warping

- At a sharp corner or branch in the cross-section, the warping function is continuous along the contour ($p = 0$), but is not continuous for other values of p .
- The warping function is not exact in the sense of the St. Venant warping function. The St. Venant warping function satisfies

$$\int (\phi_{,s}^2 + \phi_{,t}^2 - \hat{t}\phi_{,s} + \hat{s}\phi_{,t}) dA = 0 \quad (2.4-65)$$

but the thin-walled warping function used here has

$$\int (\phi_{,s}^2 + \phi_{,t}^2 - \hat{t}\phi_{,s} + \hat{s}\phi_{,t}) dA = \int 2p^2 dA \quad (2.4-66)$$

- For additional insight into the torsional response described in Section 2.4.3.5, consider the torsional response under elastic conditions. The bishear can then be written

$$\begin{aligned} F_b &= G\theta_{r,r} \int ((\phi_{,s} - \hat{t})\phi_{,s} + (\phi_{,t} + \hat{s})\phi_{,t}) dA \\ &= G\theta_{r,r} \int (\phi_{,s}^2 + \phi_{,t}^2 - \hat{t}\phi_{,s} + \hat{s}\phi_{,t}) dA \end{aligned} \quad (2.4-67)$$

If we had used the exact St Venant warping function, this term would be equal to zero, implying that the external torque in a beam free to warp is equilibrated entirely by the torque calculated from the shear stresses.

But because we use the approximate warping function given above, the bishear is nonzero. It turns out that, in an open thin-walled section, the bishear and torque calculated from the shear stresses can be written

$$F_b = G\theta_{r,r} \int 2p^2 dA, \quad M_r = G\theta_{r,r} \int 2p^2 dA \quad (2.4-68a,b)$$

so that each term equilibrates one-half of the external torque. The torsional rigidity can then be observed from the expression

$$T = F_b + M_r = G\theta_{r,r} \int 4p^2 dA = GJ\theta_{r,r} \quad (2.4-69)$$

implying $J = \int 4p^2 dA$. For a cross-section with constant thickness a and length b , this formula gives $J = \frac{1}{3}a^3b$, which is the textbook solution for the torsional rigidity of a thin plate.

Closed thin-walled sections

We consider here only closed sections with just one closed circuit in the contour. It is assumed that the shear strain flow $\gamma_{rq}b$ is equal to $K\theta_{r,r}$ on the contour, where b is the cross-section thickness and K is a constant to be determined. Thus

$$u_{r,q} + u_{q,r} = \frac{K}{b}\theta_{r,r} \quad (2.4-70)$$

As in the open cross-section theory, $u_q = \hat{p}\theta_r$, thus

$$u_r|_{(0,q)} = \left(\int_0^q \left(\frac{K}{b} - \hat{p} \right) dq \right) \theta_{r,r} \quad (2.4-71)$$

The constant K is determined by noting that the change in warping displacement resulting from one complete circuit must be zero, therefore

$$K = \frac{\oint \hat{p} dq}{\oint \frac{1}{b} dq} = \frac{2A}{\oint \frac{1}{b} dq} \quad (2.4-72)$$

where A is the area enclosed by the contour.

Using $\gamma_{rp} = 0$ along lines perpendicular to the contour, the warping displacement is

$$u_r|_{(p,q)} = \left(\int_0^q \left(\frac{K}{b} - \hat{p} \right) dq + p \hat{q} \right) \theta_{r,r} \quad (2.4-73)$$

and the warping function is

$$\phi(s,t) = \int_0^q \left(\frac{K}{b} - \hat{p} \right) dq + p \hat{q} \quad (2.4-74)$$

Again the total warping is composed of two contributions, the contour warping and the thickness warping.

Open and closed thin-walled sections

Up to now, we have not considered the location of pole P relative to the origin. And we have also not considered the origin of the p,q coordinate system. Different choices of contour origin and pole origin give different warping functions.

However, we fix the contour and pole origin using the conditions (2.4-2) and (2.4-3) and it turns out that the pole origin fixed using these conditions is the shear center of the cross-section.

It may be of interest to observe that the choice of contour and pole origins does not affect the torsional rigidity

$$J = \int \left((\phi_{,s} - \hat{t})^2 + (\phi_{,t} + \hat{s})^2 \right) dA; \text{ however the warping constant}$$

$I_\omega = \int \phi^2 dA$ is minimized by fixing the contour and pole origin using conditions (2.4-2) and (2.4-3).

2.4.4 Beam interpolations

2.4.4.1 Standard beam

The displacements and rotations are interpolated from the nodal displacements and rotations using

$$\begin{aligned} u_0 &= L_1 u^1 + L_2 u^2 \\ v_p &= H_1 v_p^1 + H_2 \theta_t^1 + H_3 v_p^2 + H_4 \theta_t^2 \\ w_p &= H_1 w_p^1 - H_2 \theta_s^1 + H_3 w_p^2 - H_4 \theta_s^2 \end{aligned} \quad (2.4-75)$$

and

$$\begin{aligned} \theta_r &= L_1 \theta_r^1 + L_2 \theta_r^2 \\ \theta_s &= -H_{1,r} w_p^1 + H_{2,r} \theta_s^1 - H_{3,r} w_p^2 + H_{4,r} \theta_s^2 \\ \theta_t &= H_{1,r} v_p^1 + H_{2,r} \theta_t^1 + H_{3,r} v_p^2 + H_{4,r} \theta_t^2 \end{aligned} \quad (2.4-76)$$

where the nodal displacements and rotations are

$(u^1, v_p^1, w_p^1, \theta_r^1, \theta_s^1, \theta_t^1)$ for node 1 and $(u^2, v_p^2, w_p^2, \theta_r^2, \theta_s^2, \theta_t^2)$ for node 2, and in which

$$L_1 = 1 - \frac{r}{L}, \quad L_2 = \frac{r}{L} \quad (2.4-77)$$

are the linear interpolation functions and

$$\begin{aligned} H_1 &= 1 - 3 \frac{r^2}{L^2} + 2 \frac{r^3}{L^3}, \quad H_2 = r - 2 \frac{r^2}{L} + \frac{r^3}{L^2}, \\ H_3 &= 3 \frac{r^2}{L^2} - 2 \frac{r^3}{L^3}, \quad H_4 = -\frac{r^2}{L} + \frac{r^3}{L^2} \end{aligned} \quad (2.4-78)$$

are the cubic interpolation functions (Hermitian displacement functions).

- Notice that the Euler-Bernoulli condition (2.4-20) is satisfied by the choice (2.4-75) and (2.4-76).
- It is seen that the transverse shear center axis displacements v_p and w_p are cubic and that the axial displacement u_o and torsional rotation θ_r are linear.
- We refer the transverse displacements to the neutral axis using (2.4-18):

$$\begin{aligned} v_p^1 &= v_n^1 - c_t \theta_r^1 & v_p^2 &= v_n^2 - c_t \theta_r^2 \\ w_p^1 &= w_n^1 + c_s \theta_r^1 & w_p^2 &= w_n^2 + c_s \theta_r^2 \end{aligned} \quad (2.4-79)$$

and we can then write the nodal degrees of freedom as $(u^1, v_n^1, w_n^1, \theta_r^1, \theta_s^1, \theta_t^1)$ for node 1 and $(u^2, v_n^2, w_n^2, \theta_r^2, \theta_s^2, \theta_t^2)$ for node 2. In this way, all nodal quantities are referred to the neutral axis. The forces and moments at the beam element nodes (energetically conjugate to the nodal displacements) are also all referred to the neutral axis.

- It should be clear from the above that warping effects are included in the standard beam. It should also be clear that pure torsion will cause the transverse displacements at the neutral axis to be non-zero, due to the shear center offsets (c_s, c_t) . And, conversely, transverse forces will cause torsion.

2.4.4.2 Warping beam

The axial rotation is interpolated using a cubic interpolation function and the axial rotation and warping degrees of freedom:

$$\theta_r = H_1 \theta_r^1 + H_2 \alpha^1 + H_3 \theta_r^2 + H_4 \alpha^2 \quad (2.4-80)$$

Evaluating $\theta_{r,r}$ at nodes 1 and 2 gives the result $\theta_{r,r}|_{\text{node 1}} = \alpha^1$ and

similar for node 2. Hence α^1, α^2 are the warping function multipliers evaluated at nodes 1, 2.

The remaining displacements and rotations are interpolated as in (2.4-75) to (2.4-78). Thus the nodal degrees of freedom

are $(u^1, v_n^1, w_n^1, \theta_r^1, \theta_s^1, \theta_t^1, \alpha^1)$ for node 1 and

$(u^2, v_n^2, w_n^2, \theta_r^2, \theta_s^2, \theta_t^2, \alpha^2)$ for node 2. Again, the nodal quantities are referred to the neutral axis using (2.4-79).

- The force conjugate to the warping degree of freedom is the bimoment, see the discussion in Section 2.4.3.5. The dimension of the bimoment is $[\text{FL}^2]$. The bimoment is included in the global force vector.
- The warping degree of freedom can be fixed or left free, and depending upon the physical situation, it may or may not be appropriate to fix the warping degree of freedom, see discussion in Section 2.4.9.1.
- The standard beam element is not recovered by fixing the warping degrees of freedom at all nodes in the beam. This is because the interpolation of axial rotations in (2.4-80) remains cubic when the warping degrees of freedom are fixed.

2.4.4.3 Comparison of standard beam and warping beam

- The motivation for introducing the warping beam element is as follows. As seen above, warping is assumed in the standard beam element through the terms J, I_ω . However the term $\theta_{r,rr}$ is always zero in the standard beam element, by construction, so warping through term I_ω does not contribute to the strain energy (but warping through term I_ω contributes to the kinetic energy). More importantly, the warping displacements between adjacent elements are not compatible in the standard beam element. This is because the term $\phi\theta_{r,r}$, which expresses the longitudinal displacement due to warping, is constant in each element and in general different between adjacent elements.

- In the warping beam, term $\phi\theta_{r,r}$ is continuous between adjacent elements, thus this element could also be called a "compatible warping beam" element.

- Due to the incompatibility of the warping displacements in the standard beam element, the results obtained from the standard beam element will be incorrect in the general case.

- However, it should be noticed that there are some important special cases in which the standard beam gives correct results:

- 1) Warping function is zero or negligible. This occurs for circular cross-sections.

- 2) Angle of twist per unit length is constant along the length of the beam assemblage. This occurs in many static problems, e.g. bending and torsion of a straight beam. This situation also occurs in problems in which torsion is not present, for example, in-plane deformations.

- 3) Warping, although present, does not affect the solution (other than the torsional response) significantly. This occurs for solid sections and possibly also for closed thin-walled sections.

- In general, for the solid and closed thin-walled sections, the standard beam can be used without significant error. However for the open thin-walled sections, the warping beam should be used. It is allowed, however, to use the open thin-walled sections with the standard beam, and, as long as the angle of twist per unit length is constant along the beam, the warping beam and standard beam will give exactly the same results.

2.4.5 Beam element implementation

- The stiffness matrix, mass matrix and force vector are formulated in the local degrees of freedom (in the r , s , t axes). These matrices and vectors are then transformed to the nodal degrees of freedom (either global or skew) and assembled into the global system matrices.

2.4.5.1 Linear formulation

- It is assumed that the displacements, rotations, and strains are infinitesimally small, and the elastic-isotropic material is used. The Wagner effect is not included.

2.4.5.2 Materially-nonlinear-only formulation

- It is assumed that the displacements, rotations, and strains are infinitesimally small. Either the elastic-isotropic material is used (for example, in conjunction with element birth-death), the plastic material models are used, or the moment-curvature material model is used. The Wagner effect is not included.

2.4.5.3 Large displacement formulation

- It is assumed that large displacements/rotations can occur, but only small strains. Any beam material model, linear or nonlinear, can be used. The input of beam cross-sections and material data is exactly the same as when using the linear or materially-nonlinear-only beam elements.
- If a stressed large displacement beam element is subjected to a rigid body rotation, the stresses/forces (expressed in the local coordinate system) do not change during the rigid body rotation.
- The geometry of the large displacement beam element is shown in Fig 2.4-9. The shape of the beam neutral axis is completely specified by the positions of the end-nodes, and by the orientation of the end-node triads. Each end-node triad consists of three orthogonal unit vectors $\mathbf{V}_r, \mathbf{V}_s, \mathbf{V}_t$. Initially the triads are identical to the element coordinate axes unit vectors, e.g, \mathbf{V}_r is a unit vector in the r direction, etc.

During the deformations, the positions of the end-nodes are updated, as usual, by the nodal displacements. The orientations of the end-node triads are updated incrementally by the increments in nodal rotations. The end-node triads can rotate independently of

each other, however it is assumed that the relative rotation of the end-node triads remains small.

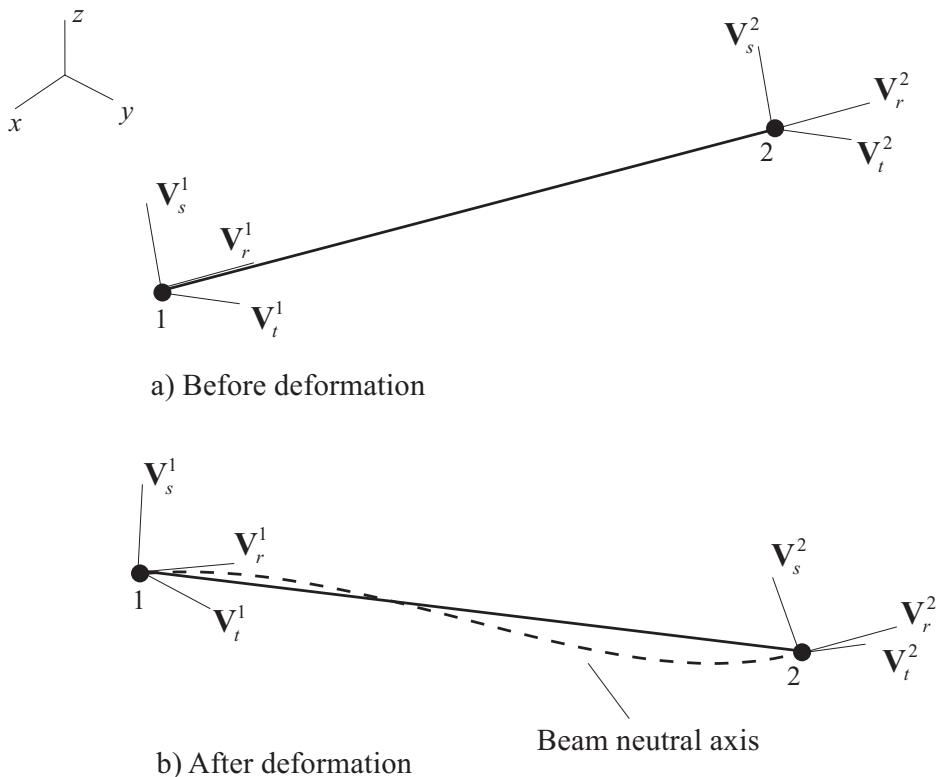


Figure 2.4-9: Geometry of large displacement beam element

- In the large displacement formulation, the coordinate system in which the beam local displacements and rotations are measured is updated during the solution. In ADINA 8.8 and higher, this coordinate system is taken from the orientations of the end-node triads. In ADINA 8.7 and lower, this coordinate system is taken from the coordinates of the end-nodes. Hence the solution output in ADINA 8.8 and higher will in general be different than the solution output in ADINA 8.7 and lower.

The advantage of using a coordinate system defined from the orientations of the end-node triads is that this coordinate system is uniquely defined even for general 3-D deformations including

torsion. A coordinate system defined from the end-nodes coordinates is not uniquely defined for 3-D deformations including torsion.

- The shape of the beam neutral axis becomes a curved space curve. The displacements and rotations of the beam nodes, used in the construction of the force vector and stiffness matrix, are measured with respect to the beam local coordinate system.
- Fig 2.4-10 shows an example for in-plane bending. A single beam element is clamped at node 1 and subjected to a prescribed rotation θ at node 2. The coordinate system \tilde{x}, \tilde{y} used to measure beam local displacements and rotations is located halfway between the nodes, and is rotated an angle $\theta/2$ with respect to the global coordinate system x, y . Notice that the \tilde{x} direction does not coincide with the line between the end-nodes.

The rotation at local node 2 with respect to the local coordinate system is $\theta/2$, and the rotation at local node 1 with respect to the local coordinate system is $-\theta/2$. Also, the transverse displacement at local node 2 with respect to the local coordinate system is $-\tilde{v}$ and the transverse displacement at local node 1 with respect to the local coordinate system is \tilde{v} .

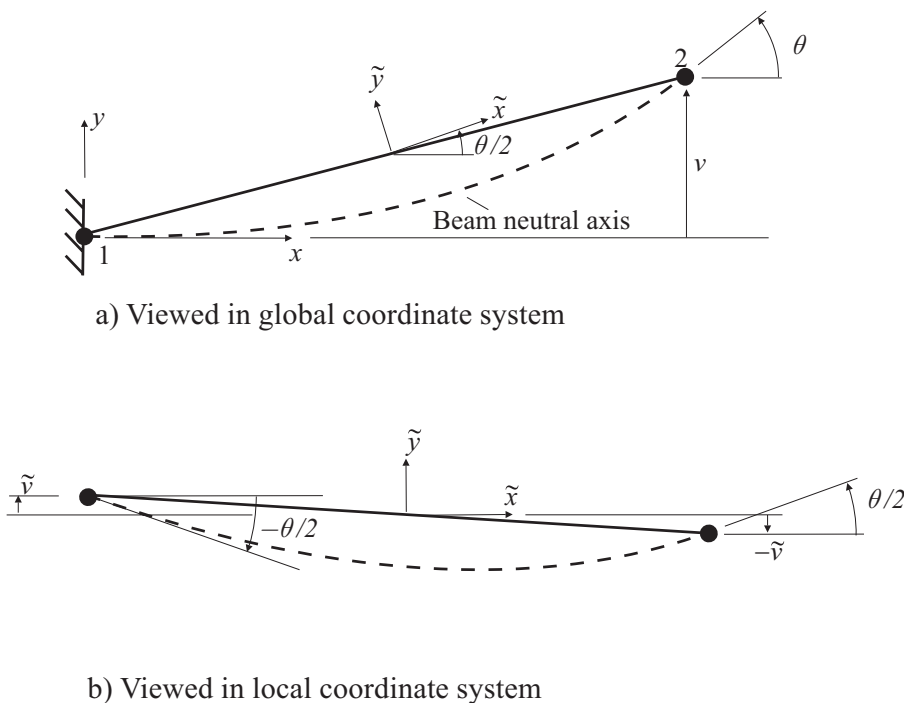


Figure 2.4-10: Coordinate system update for large displacement beam element, in-plane bending example

- The element stiffness matrix is non-symmetric in the rotational degrees of freedom. Therefore the option of using the non-symmetric sparse solver (MASTER SOLVER=NONSYS-SPARSE) can be useful in obtaining convergence.
- The Wagner effect is included by default for the thin-walled sections.
- For frequency analysis and linearized buckling analysis, it is allowed to include certain 2nd order stiffness effects into the stiffness matrix. This choice is made using the command

KINEMATICS KBEAM-EIGENVALUE=YES

By default, this feature is not used. Note that these 2nd order

stiffness effects are not introduced into the force vector, so this feature should be used with caution.

This feature is made available only for backwards compatibility with version 8.7 and earlier.

The following effects are not included within this feature: warping, shear center offsets, Wagner effect.

2.4.5.4 Mass matrices

- The beam element can be used with a lumped or a consistent mass matrix, except for explicit dynamic analysis which always uses a lumped mass matrix.
- The consistent mass matrix of the beam element is evaluated in closed form, and does not include the effect of shear deformations. The matrix is defined in the local coordinate system using the expression for the kinetic energy derived above (equation 2.4-54) and the beam interpolations (either standard or warping). Therefore the consistent mass matrix includes warping effects.

For the standard beam element, the consistent mass matrix is the same as the one given in the following reference:

ref. J.S. Przemieniecki, *Theory of Matrix Structural Analysis*, McGraw-Hill Book Co., 1968.

with the exception that there are additional terms arising from the warping constant I_ω .

- The lumped mass for translational degrees of freedom is $M/2$ where M is the total mass of the element.
- The rotational lumped mass for static analysis is 0.

The rotational lumped mass for implicit dynamic analysis is

$$M_{rr} = \frac{2}{3} \cdot \frac{M}{2} \cdot \frac{I_{ss} + I_{tt}}{A}. \text{ This lumped mass is applied to all}$$

rotational degrees of freedom in order to obtain a diagonal mass matrix in any coordinate system.

The rotational lumped mass for explicit dynamic analysis is

$$M_{rr} = 3 \cdot \frac{M}{2} \cdot \frac{I_m}{A} \text{ where } I_m = \max(I_{ss}, I_{tt}) \text{ is the maximum}$$

bending moment of inertia of the beam. This lumped mass is applied to all rotational degrees of freedom. Note that this scaling of rotational masses ensures that the rotational degrees of freedom do not affect the critical stable time step of the element.

- The rotational lumped masses can be multiplied by a user-specified multiplier ETA (for implicit dynamic analysis only). The multiplier factor used in explicit analysis is always 1.0.
- For the warping element, lumped masses for the warping degrees of freedom are defined. However the process of lumping implies that all coupling between warping and rotational degrees of freedom is lost. Since there is a high degree of coupling between these degrees of freedom, the lumped mass assumption must be considered extremely approximate. Therefore the use of a lumped mass matrix is not recommended for the warping beam element.

The lumped warping masses have been improved between versions 8.9 and 9.0, so the results of analyses in which lumped warping masses are used change between versions 8.9 and 9.0.

- In theory the consistent mass matrix, expressed in the global coordinate system, changes as a result of large rotations in large displacement analysis. However, the mass matrix is not updated (except in the case of element birth/death). Therefore, when using a consistent mass matrix in large displacement analysis, the results can be in error if the rotations are significant.

2.4.5.5 Elastic beam element

- The beam element stiffness matrix is evaluated in closed form. The stiffness matrix used is derived from the strain energy expression in equation (2.4-39) and the beam element interpolations (either standard or warping).

For the standard beam element, the stiffness matrix is the same as the one given in the following reference:

ref. J.S. Przemieniecki, *Theory of Matrix Structural Analysis*, McGraw-Hill Book Co., 1968.

- Certain cross-sections allow entry of the shear areas. When shear areas are entered, the force vector and stiffness matrix are

modified in order to approximate the effects of shear deformations, as discussed in Przemieniecki.

For example, considering a cantilever beam with in-plane deformations only, the stiffness matrix for the free transverse displacement and rotation, including the shear area modifications is

$$\mathbf{K} = \begin{bmatrix} \frac{12EI_{tt}}{L^3(1+\phi_2)} & \frac{6EI_{tt}}{L^2(1+\phi_2)} \\ sym & \frac{EI_{tt}(4+\phi_2)}{L(1+\phi_2)} \end{bmatrix}$$

where

$$\begin{aligned} \phi_2 &= \frac{12EI_{tt}}{L^2GA_s^{sh}}, \quad A_s^{sh} > 0 \\ &= 0, \quad A_s^{sh} = 0 \end{aligned}$$

- The coefficient of thermal expansion can be specified as a material property. The coefficient of thermal expansion is constant (independent of the temperature). In addition, the beam temperature is taken as the average of the temperatures of the beam end-nodes. The temperature gradients at beam nodes are ignored.

2.4.5.6 Elastic-plastic beam element

- The element is used with the plastic-cyclic material model described in Section 3.4.6. The element can also be used with a plastic-bilinear material with isotropic hardening, but then the program automatically converts the material input into the equivalent plastic-cyclic material input.

The stress-strain law used incorporates the assumptions that the stresses τ_{ss} , τ_{tt} , τ_{st} are zero.

- The material model can be used either with the materially-nonlinear-only formulation or with the large displacement formulation (in which case the displacements/rotations can be large). In all cases, the strains are assumed to be small.
- The Wagner effect is included by default when the large

displacement formulation is selected.

- ref. KJB* • All element matrices in elasto-plastic analysis are calculated using numerical integration.
Section 6.6.3

In the r -direction, Newton-Cotes integration is used (Fig 2.4-11). The number of integration points is controlled with the EGROUP BEAM ... RINT parameter. RINT can be 3, 5, 7 or 9. The default is 5, which corresponds to an exact integration of all system matrices under elastic small strain conditions (no Wagner effect). When the Wagner effect is included, 9 stations can be selected in order to exactly integrate the system matrices under elastic conditions. However, 9 stations should not be used in general as 9-point Newton-Cotes integration is possibly unreliable for the numerical integration of arbitrary functions.

The numerical integration in the cross-section depends upon the cross-section, therefore further details are given along with the cross-sections (Section 2.4.6).

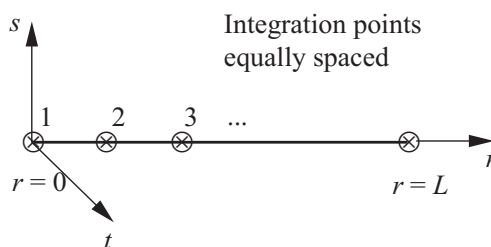


Figure 2.4-11: Integration point locations in r -direction

- Note that the system matrices are identical to the system matrices used in elastic analysis if

► The warping function assumed in elasto-plastic analysis is the same as the one used in elastic analysis. For the pipe cross-section, this condition is automatically satisfied since there is no warping function in the pipe section. For the rectangular cross-section, this condition is reached in the limit for a very thin section (because the exact warping function is employed in the limit of a very thin section). For the thin-walled sections, this condition is satisfied as long as the torsional rigidity formula used in elastic analysis is formula FW (see Section 2.4.6 for details about the formulas FW).

- ▶ The numerical integration is of high enough order.
- ▶ The material is (still) elastic.
- For the rectangular and pipe cross-sections, the elastic-plastic beam element can either be used using a 2-D action option or a 3-D action option. This choice is made using the command `EGROUP BEAM SUBTYPE={TWO-D/THREE-D}`; the default is `THREE-D`.
 - 2-D action assumes that the element deformations occur in the r - s plane. In this case, the element has no stiffness corresponding to deformations out of the r - s plane. Also, the element does not have any torsional stiffness.
 - 3-D action assumes that element deformations occur in any direction.
- For both 2-D action and 3-D action, the element can have an arbitrary orientation with respect to the global coordinate axes.
- For the box, I, U, L cross-sections, the elastic-plastic beam element does not include shear deformation effects.

2.4.5.7 Element output

- There are three options for calculation of beam element output:
 - ▶ element nodal point forces/moments (`EGROUP BEAM RESULTS=FORCES`) (the default for elastic analysis)
 - ▶ stresses at the element integration points (`EGROUP BEAM RESULTS=STRESSES`) (the default for elastic-plastic analysis)
 - ▶ element section forces/moments at discrete locations along the axial direction of the element (`EGROUP BEAM RESULTS=SFORCES`)

In the description below, we use the term "forces" to describe both forces and moments.

Nodal point forces: The nodal point forces are computed at the element local nodes. These forces are equivalent, in the virtual work sense, to the internal element stresses. The nodal point forces are used to calculate reaction forces.

- Because the local nodes are located at the neutral axis (and not the shear center), the nodal point forces are also computed with reference to the neutral axis (and not the shear center). This issue is discussed in further detail below.
- The element nodal point forces are accessible in the AUI using the variable names `NODAL_FORCE-R`, `NODAL_FORCE-S`, `NODAL_FORCE-T`, `NODAL_MOMENT-R`, `NODAL_MOMENT-S` and `NODAL_MOMENT-T`. (If the warping beam is used, the variable name `NODAL_BIMOMENT` is also available.) In the AUI, element local nodes are defined as element points of type label. For example, to access the result computed at element 5, local node 2, define an element point of type label with element number 5, label number 2.
- If nodal point forces are requested in an elastic-plastic analysis, the state of stress in the element (whether elastic or plastic) is not indicated.

Stresses: Stress output is not available for the elastic beam element. If stresses are requested for the elastic element, the nodal point forces are computed instead.

Stress output for the elastic-plastic beam element is described in Section 3.4.5.3.

Section forces: The element section forces can be computed at equally spaced points along the neutral axis of the beam. The number of points is controlled by `EGROUP BEAM ... SPOINT`; the default number of points is 2. Station 1 is at local node 1 and station `SPOINT` is at local node 2. The sign convention for section forces is given in Figure 2.4-12. Notice that the vector of positive moment in the *s*-direction points in the negative *s*-direction.

If section forces are requested in an elastic-plastic analysis, the state of stress in the element (whether elastic or plastic) is not indicated.

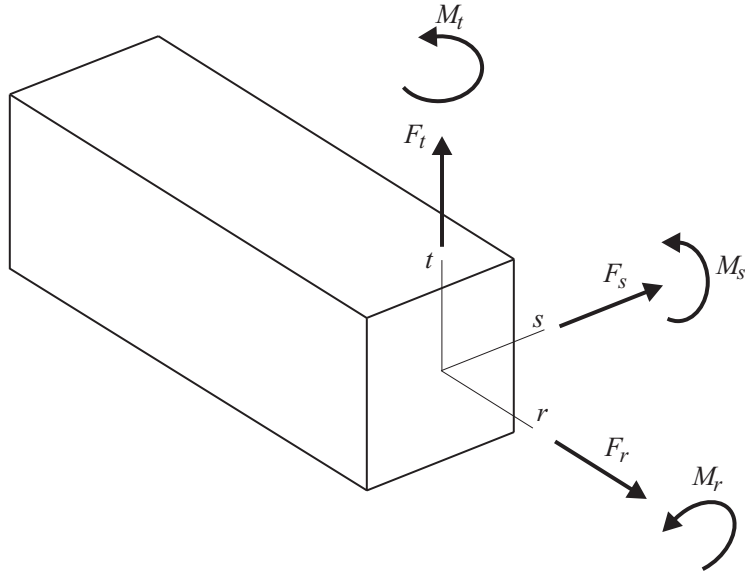


Figure 2.4-12: Sign convention for section forces/moments

There are two options for computation of section forces/moments.

Integration of stresses over the cross-section

The option of integration of stresses over the cross-section can be used for the elastic and elastic-plastic beam elements, for geometrically linear and large displacement formulations. The section forces are computed using

$$\begin{aligned}
 F_r &= \int \tau_{rr} dA \\
 M_r &= \left(\int (\hat{s}^2 + \hat{t}^2) \theta_{r,s} \tau_{rr} + (\phi_{s,s} - \hat{t}) \tau_{rs} + (\phi_{s,t} + \hat{s}) \tau_{rt} \right) dA \\
 M_s &= \int -t \tau_{rr} dA \\
 M_t &= \int -s \tau_{rr} dA \\
 M_b &= \int \phi \tau_{rr} dA
 \end{aligned} \tag{2.4-81}$$

These expressions are developed from the principle of virtual work:

F_r, M_r, M_s, M_t, M_b are conjugate to $\delta u_{o,r}, \delta \theta_{r,r}, -\delta \theta_{s,r}, -\delta \theta_{t,r}, \delta \theta_{r,rr}$ respectively. In M_r , the term involving τ_{rr} arises from the Wagner effect and is included only when the Wagner effect is included. M_b is computed only for the warping beam.

The shear forces are estimated using the expressions $F_s = -M_{t,r}, F_t = -M_{s,r}$. (It is not possible to compute shear forces by integration of shear stresses because in Euler-Bernoulli beam theory, there are no shear stresses generated except in torsion.)

For the elastic beam, the section forces are evaluated analytically. For the elastic-plastic beam, the axial force and moments are evaluated using numerical integration over the cross-section, and the shear forces are evaluated by the approximations

$$F_s = -\frac{M_t|_{r=1} - M_t|_{r=0}}{L}, \quad F_t = -\frac{M_s|_{r=1} - M_s|_{r=0}}{L}.$$

For the elastic-plastic beam, it is necessary that each station at which section forces are evaluated correspond to an integration point along the r direction. Thus the expression $\frac{\text{RINT} - 1}{\text{SPOINT} - 1}$ must be an integer for the elastic-plastic beam.

Fixed-end force correction

As an alternative for computing section forces by integration, it is allowed to compute section forces using the theory of fixed-end force correction.

In order to motivate the fixed-end force correction, consider the problem shown in Fig. 2.4-13. Only one element is required to obtain the analytical displacement and rotation solution. However, the shear forces and bending moments, computed without using the fixed-end force correction, are incorrect, and only approach the analytical solutions as more elements are used.

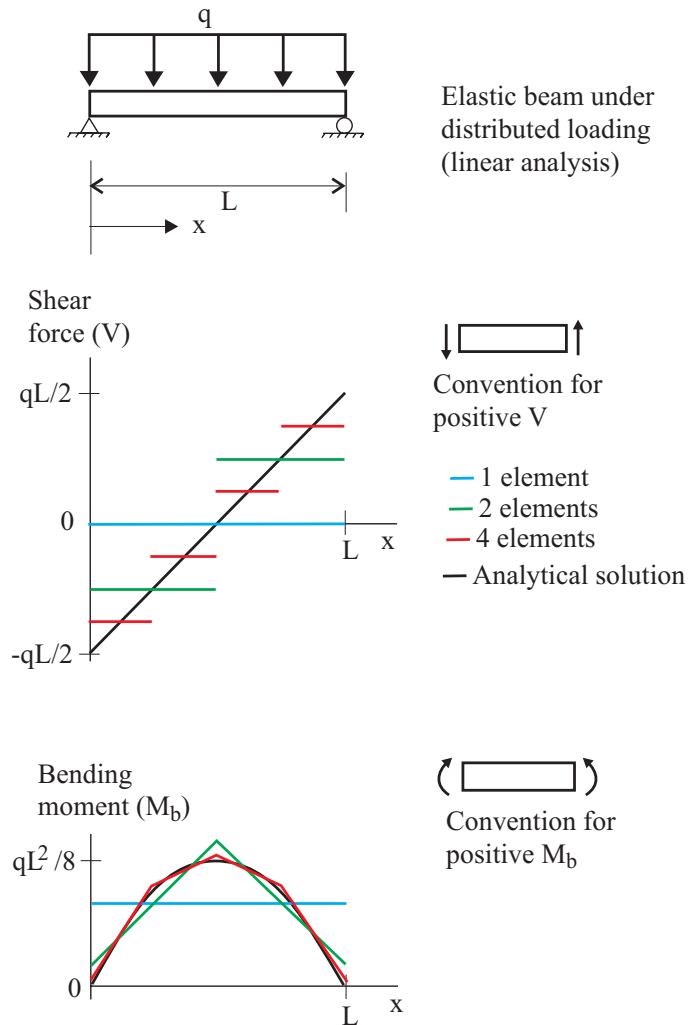


Figure 2.4-13: Schematic beam element forces/moments when subjected to distributed loading

The basic ideas of fixed-end force correction are shown in Fig. 2.4-14. As seen, the original distributed loading in Fig. 2.4-14(a) is replaced by the equivalent nodal point forces and moments in Fig. 2.4-14(b). (Of course, this procedure is the one always used in finite element analysis when distributed loads are applied.) The solution without any fixed-end force correction is shown in Fig. 2.4-14(c) and it is clear that the bending moment distribution does not agree with the results from beam theory. This bending moment distribution arises from the internal force vector in the beam element, namely the beam element is subjected to the end-node forces and moments and equilibrium is used to determine the intermediate section forces and moments. In the fixed-end force correction, the beam element is subjected, instead, to: 1) the end-node forces and moments, from which the nodal point equivalent forces of the distributed load have been removed, and 2) the original distributed load. Then equilibrium is used to determine the intermediate section forces and moments. The result is shown in Fig. 2.4-14(d), and the bending moment diagram corresponds to beam theory.

Similarly, when the fixed-end force correction is used for the problem in Fig. 2.4-13, the exact shear forces and bending moments are obtained for each finite element mesh.

The fixed-end force correction feature does not account for body force loads acting on the element, for example, from mass-proportional loads.

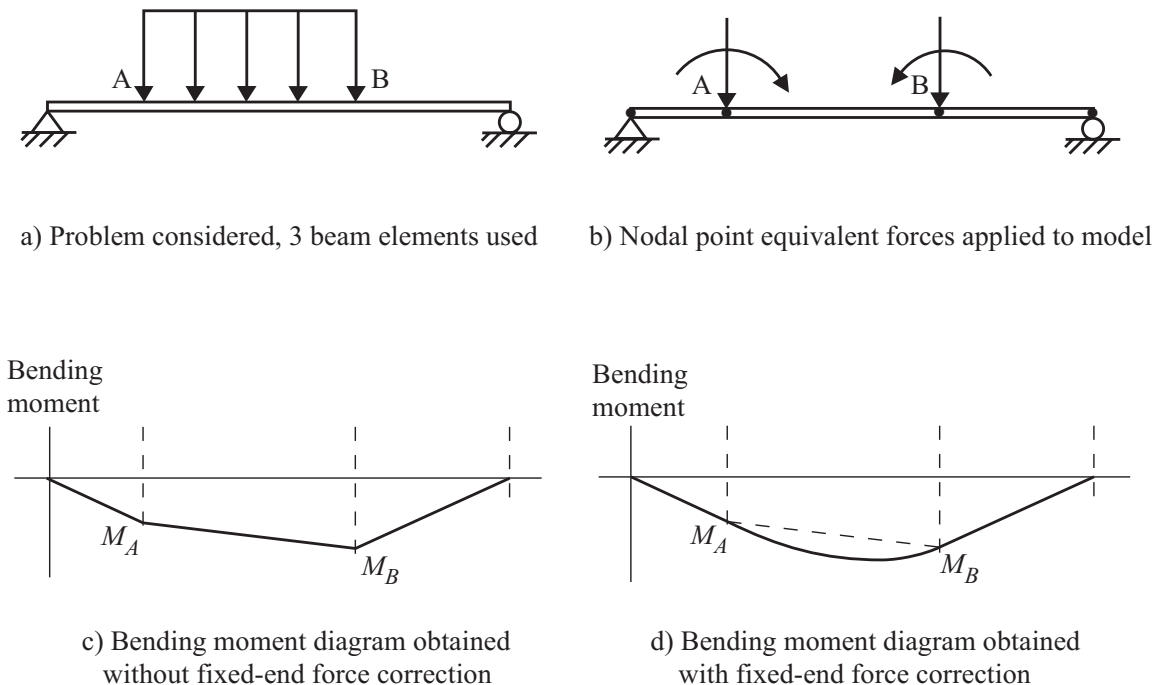


Figure 2.4-14: Fixed-end force correction example

- For more information on the fixed end force correction, see the following references:

ref: W. McGuire and R.H. Gallagher, *Matrix Structural Analysis*, John Wiley & Sons, New York, 1979.

ref: M.L. Bucelem and K.J. Bathe, *The Mechanics of Solids and Structures— Hierarchical Modeling and the Finite Element Solution*, Springer, 2011.

- The option of fixed-end force correction is selected using MASTER FEFCORR=YES (not the default).

The option of fixed-end force correction is applicable only to linear static analysis. In addition the fixed-end force correction cannot be used when the beam element uses the rigid end feature described in Section 2.4.8.

If the fixed-end force correction is requested for an element

group in which the fixed-end force correction cannot be performed, the section forces are computed by integration of stresses over the cross-section.

Section forces in the AUI

The section forces are accessible in the AUI using the variable names `AXIAL_FORCE`, `BENDING_MOMENT-S`, `BENDING_MOMENT-T`, `SHEAR_FORCE-S`, `SHEAR_FORCE-T` and `TORSIONAL_MOMENT`. When the warping beam element is used, the variable `BIMOMENT` is also available.

Comparison of nodal point forces and section forces

Consider the cantilever beam shown in Fig. 2.4-15. The section twists under the transverse load, because the transverse load is applied at the neutral axis, and not at the shear center.

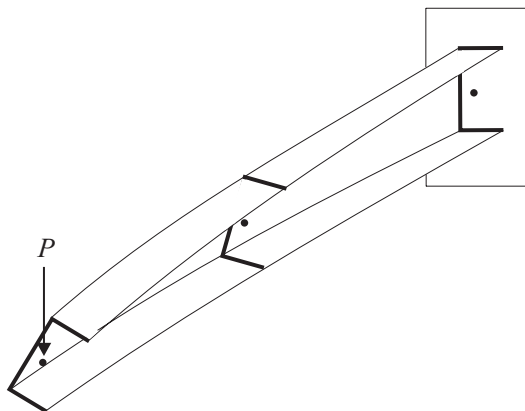


Figure 2.4-15: Twisting of beam due to transverse force on neutral axis

The nodal point moment in the r -direction is zero for this problem. This result is consistent with the reaction at the fixed end, in which the moment reaction along the beam direction is also zero.

However the section torsional moment computed using (2.4-81) is non-zero. Near the built-in end, where the warping is fixed, the torque produced by the load about the shear center is equilibrated

by the change in bimoment and the section torsional moment is nearly zero (see equation 2.4-34). Far from the built-in end, the section torsional moment equals the torque produced by the load P .

Thus for problems in which the neutral axis and shear center do not coincide, the section forces can be more "physically realistic" than the nodal point forces.

- When the neutral axis and shear center coincide, then, in linear elastic analysis, the nodal point forces and moments are equal to the section forces and moments computed by integration over the cross-section. The nodal point transverse shear forces are also equal to the section shear forces.
- When the neutral axis and shear center coincide, then, when the element undergoes plasticity in elastic-plastic analysis, the nodal point forces and moments are not in general equal to the section forces and moments computed by integration over the cross-section. Then, for example, the reaction forces might not exactly agree with the corresponding section forces. However, as the mesh is refined, the nodal point forces and moments will more closely match the section forces and moments.

The shear forces are generally less accurate than the bending moments, because numerical differentiation is used in the computation of the shear forces.

2.4.6 Cross-sections

The following cross-section types are available:

General (properties)
Rectangular
Pipe
Box
I
U
L

2.4.6.1 General cross-section

- The section area, moments of inertia, and other properties can

be directly specified using the CROSS-SECTION PROPERTIES command. In this case, the formulas given in Section 2.4.2 should be used to calculate these properties.

- The general cross-section can be used in both linear elastic and large displacement elastic analysis.
- The general cross-section can be used with either the standard beam element or the warping beam element. The choice of beam element depends on the values of the cross-section geometric properties, for example, if the torsional rigidity is much lower than the polar moment of inertia, the section behaves similarly to a thin-walled open section and the warping beam element should be used.
- In version 8.9 and lower, when the general cross-section is used with the standard beam, the offsets to the shear center are ignored and the principal directions are aligned with the s - t axes (that is, parameters CTOFFSET, CSOFFSET, STINERTIA in the CROSS-SECTION PROPERTIES command are ignored). This restriction is removed in version 9.0 and higher. Therefore when running models created in version 8.9 and lower, it may be necessary to set CTOFFSET, CSOFFSET, STINERTIA to zero in order to obtain the same results as in version 8.9.
- To include the Wagner effect of torsional coupling in large displacements, set RRINERTIA, SRINERTIA, TRINERTIA, WRINERTIA, DRINERTIA $\neq 0$ in the CROSS-SECTION PROPERTIES command.
- In version 8.9 and lower, terms SRINERTIA, TRINERTIA, WRINERTIA, DRINERTIA are defined slightly differently than in version 9.0 and higher. In addition, term RRINERTIA is not present in version 8.9 and lower. Therefore, when converting data used in version 8.9 and lower, in which the Wagner effect is included using these terms, it is necessary to recalculate these terms, and add term RRINERTIA.

2.4.6.2 Rectangular cross-section

Figure 2.4-16 shows the rectangular cross-section and its dimensions. ADINA evaluates the section properties from the

given dimensions.

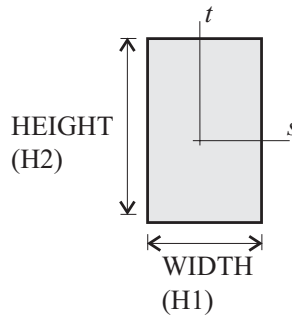


Figure 2.4-16: Rectangular cross-section

- The rectangular cross-section can be used in both linear elastic and large displacement elastic analysis. In addition, the rectangular cross-section can be used in MNO and large displacement elastic-plastic analysis.

General comments

- The shear center of the cross-section coincides with the neutral axis.
- The warping constant I_ω is assumed equal to zero.
- The Wagner effect cannot be included when using a rectangular cross-section.

Comments for elastic analysis

- The formula

$$\begin{aligned}
 J &= \frac{1}{3} \left(1 - 0.63 \frac{H1}{H2} \left(1 - \frac{H1^4}{12 H2^4} \right) \right) H1^3 H2, \quad H1 \leq H2 \\
 &= \frac{1}{3} \left(1 - 0.63 \frac{H2}{H1} \left(1 - \frac{H2^4}{12 H1^4} \right) \right) H2^3 H1, \quad H2 \leq H1
 \end{aligned}$$

is used to calculate the St. Venant torsional constant. J can be multiplied by the user-specified parameter TORFAC.

- Parameters SC, TC are used to adjust the bending moment inertias I_{ss} , I_{tt} of the cross-section using the parallel axis theorem:

$$\begin{aligned} I_{ss} &\leftarrow I_{ss} + A \times TC^2 \\ I_{tt} &\leftarrow I_{tt} + A \times SC^2 \end{aligned}$$

The centroid of the beam cross-section remains at the origin of the (s, t) system, so SC, TC cannot be interpreted as cross-section offsets. If cross-section offsets need to be modeled, use rigid links as described in Section 2.4.9.

- Parameters SSHEARF, TSHEARF can be used to calculate the effective shear cross-section areas in the s and t directions:

$$A_s^{sh} = SSHEARF \times A, A_t^{sh} = TSHEARF \times A$$

Comments for elastic-plastic analysis

- Shear deformation effects can be included in an approximate manner as follows. Additional terms are added to the beam element displacements:

$$\begin{aligned} u &= \dots + \left(-1 + 6\frac{r}{L} - 6\frac{r^2}{L^2} \right) t \beta_1 + \left(-1 + 6\frac{r}{L} - 6\frac{r^2}{L^2} \right) s \beta_2 \\ v &= \dots + \left(3\frac{r^2}{L^2} - 2\frac{r^3}{L^3} \right) \beta_2 \\ w &= \dots - \left(3\frac{r^2}{L^2} - 2\frac{r^3}{L^3} \right) \beta_1 \end{aligned}$$

where β_1, β_2 are additional degrees of freedom. These additional terms result in constant shear distortions γ_{rs} and γ_{rt} along the length of the beam, as shown in Fig. 2.4-17. The additional degrees of freedom are condensed out at the element level.

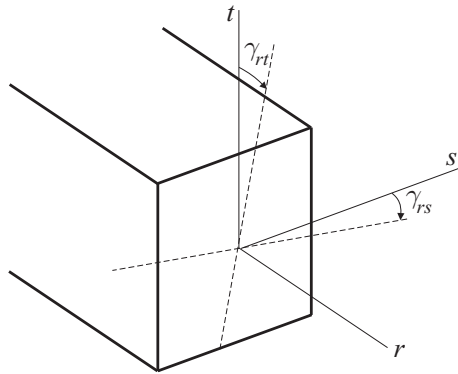


Figure 2.4-17: Assumptions of shear deformations through element thickness for elasto-plastic beam element, rectangular section

The shear effects are included using the command CROSS-SECTION RECTANGULAR ... ISHEAR=YES. The default is to not include shear effects.

- For beams in 3D action, warping effects are included by modifying the longitudinal displacement as follows:

$$u = \dots + st\alpha_1 + (s^3t - st^3)\alpha_2$$

where α_1, α_2 are additional degrees of freedom. The additional degrees of freedom are condensed out at the element level.

Warping effects are always included for beams in 3D action.

This procedure is described in the following reference:

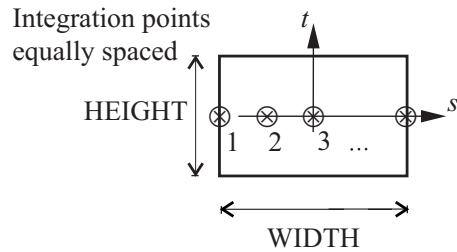
- ref. K.J. Bathe and A. Chaudhary, "On the Displacement Formulation of Torsion of Shafts with Rectangular Cross-Sections", *Int. Num. Meth. in Eng.*, Vol. 18, pp. 1565-1568, 1982.

- Table 2.4-2 and Figure 2.4-18 give the integration orders used in the cross-section.

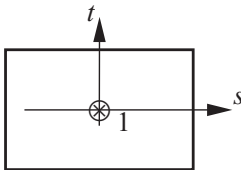
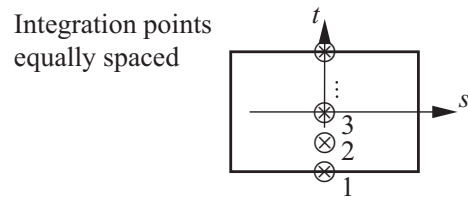
Table 2.4-2: Integration orders in elasto-plastic beam analysis,
rectangular cross-section

Coordinate	Action	Integration orders	
		Default	Maximum
s	2-D action	3	7
	3-D action	7	7
t	2-D action	1	1
	3-D action	7	7

- 1) Newton-Cotes integration is used in all coordinate directions
- 2) One point integration is used in the t -direction for the rectangular section in 2-D action regardless of user input.
- 3) Seven point integration is used in the s - and t - directions for the rectangular section in 3-D action, regardless of user input.



a) Integration point locations in s-direction

b) Integration point locations in t-direction,
2-D actionc) Integration point locations in t-direction,
3-D actionFigure 2.4-18: Integration point locations in elasto-plastic
beam analysis, rectangular cross-section

- In explicit dynamic analysis, the use of condensed degrees of freedom (shear deformations and warpings) will cause the program to run considerably more slowly than if this option were not used.

2.4.6.3 Pipe cross-section

Figure 2.4-19 shows the pipe cross-section and its dimensions. ADINA evaluates the section properties from the given dimensions.

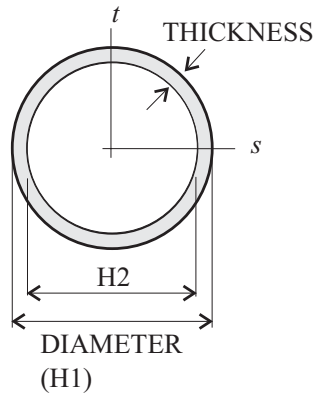


Figure 2.4-19: Pipe cross-section

- The pipe cross-section can be used in both linear elastic and large displacement elastic analysis. In addition, the pipe cross-section can be used in MNO and large displacement elastic-plastic analysis.

General comments

- The shear center of the cross-section coincides with the neutral axis.
- The warping constant I_ω is set to zero (as the pipe section cannot warp).
- The Wagner effect cannot be included when using a pipe cross-section.

Comments for elastic analysis

- The St. Venant torsional constant J is equal to the polar moment of inertia of the section, multiplied by the user-specified parameter TORFAC.
- Parameters SC, TC are used to adjust the bending moment inertias I_{ss} , I_{tt} of the cross-section, as in the rectangular cross-section.

- Parameters SSHEARF, TSHEARF can be used to calculate the effective shear cross-section areas in the s and t directions, as in the rectangular cross-section.

Comments for plastic analysis

Table 2.4-3 and Figure 2.4-20 gives the integration orders used in the cross-section. For the pipe section, polar coordinates are used:

$$R = \sqrt{s^2 + t^2}, \quad \psi = \tan^{-1} \frac{t}{s}.$$

Table 2.4-3: Integration orders in elasto-plastic beam analysis, pipe section

Coordinate	Action	Integration scheme	Integration orders	
			Default	Maximum
radius R	Any	Newton-Cotes.	3	7
tangential angle ψ	2-D action	Composite trapezoidal rule	5	5
	3-D action		8	8

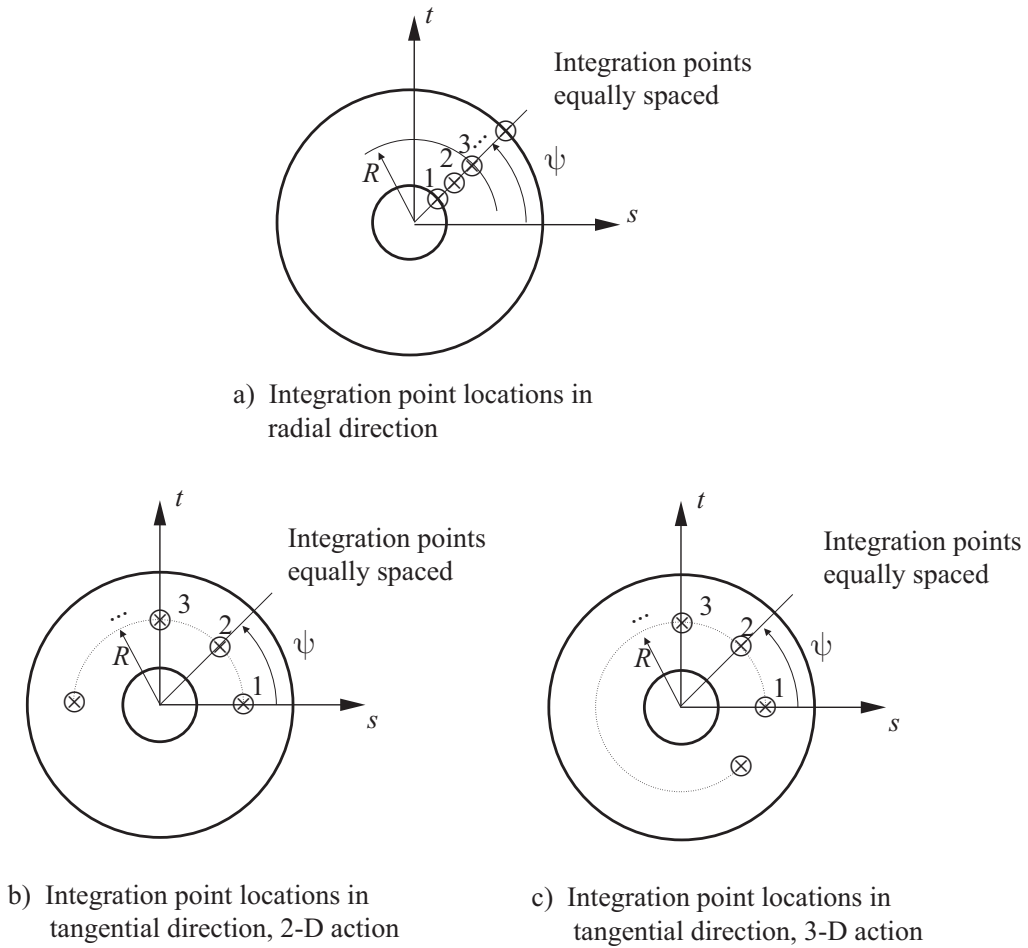


Figure 2.4-20: Integration point locations in elasto-plastic beam analysis, pipe section

2.4.6.4 Box cross-section

Figure 2.4-21 shows the box cross-section and its dimensions. ADINA evaluates the section properties from the given dimensions.

Table 2.4-4: Options for the box cross-section

	Standard beam, V89 assumptions ¹	Standard beam	Warping beam
Formula for calculation of torsional rigidity (elastic material)	F1, FW, default F1	F1, FW, default FW	FW
Warping constant I_{ω} included	No	Yes	Yes
Wagner effect allowed under large displacement conditions	No	Yes, by default	Yes, by default
Plastic material model allowed	No	Yes	Yes

1) CROSS-SECTION BOX ... ASSUMPTIONS=V89

The table shows that either the standard beam or the warping beam can be used in conjunction with the box cross-section.

- When the standard beam is used, there are two choices:

Standard beam, ASSUMPTIONS=V89:

- Warping effects ignored except for the torsional rigidity; mass matrix effects due to the warping constant I_{ω} are ignored.
- Torsional rigidity J calculated by formula F1 by default.
- Wagner effect not included in large displacement analysis.

Standard beam, ASSUMPTIONS=CURRENT:

- Warping effects included; the torsional rigidity includes warping effects and mass matrix effects due to the warping constant I_{ω} are included.
- Torsional rigidity J calculated by formula FW by default.
- Wagner effect included by default in large displacement analysis.

General comments

- The shear center of the cross-section coincides with the neutral axis.
- The calculation of I_ω is made to be consistent with the calculations done in the plastic material (in which all integrals are evaluated numerically from the warping function). Hence the value of I_ω might not be the same as the handbook values, particularly if the walls of the cross-section are thick.
- The Wagner effect in large displacements is controlled with CROSS-SECTION BOX ... WAGNER={DEFAULT, YES, NO}. DEFAULT=YES unless the standard beam element is used and ASSUMPTIONS=V89.

Comments for elastic analysis

- The torsional rigidity is controlled with CROSS-SECTION BOX ... TORFORMULA={DEFAULT, F1, FW}.

TORFORMULA=F1 (textbook formula):

$$J = K H_7 H_8$$

TORFORMULA=FW (formula consistent with assumed warping function):

$$J = \frac{1}{3} H_3^3 H_6 + \frac{1}{3} H_4^3 H_1 + K^2 \left(\frac{H_6}{H_3} + \frac{H_4}{H_1} \right)$$

TORFORMULA=DEFAULT: Use formula FW unless the standard beam is used with ASSUMPTIONS=V89, otherwise use formula F1.

In these formulas, $K = \frac{H_3 H_4 H_7 H_8}{H_3 H_7 + H_4 H_8}$ (the shear flow constant). Both of these formulas assume that the walls of the box section are thin, and then these formulas are numerically almost

equal to each other.

When the warping beam is used, formula FW is always used.

J can be multiplied by the user-specified parameter TORFAC.

- Parameters SC, TC are used to adjust the bending moment inertias I_{ss} , I_{tt} of the cross-section, as in the rectangular cross-section.
- Parameters SSHEARF, TSHEARF can be used to calculate the effective shear cross-section areas in the s and t directions, as in the rectangular cross-section.

Comments for elastic-plastic analysis

- The locations and the labeling of the integration points are given in Fig. 2.4-22. For each direction in the cross-section, Newton-Cotes integration is used with 1, 3, 5, 7 integration points in that direction (default 3). The default integration order is sufficient to exactly integrate all quantities in elastic analysis (not considering the Wagner effect). However, the integration order can be increased in order to capture the spread of plasticity throughout the cross-section.

Each integration point is labeled with a four digit number, the first digit giving the r integration point and the last three digits giving the location within the cross-section. For example, integration point 2312 denotes integration point 2 in the r direction and integration point 312 in the cross-section. This integration point is in the left web of the cross-section.

It can happen that there are two integration points at the same physical location, yet the stresses are different at these points. This happens because the warping function is not continuous at sharp corners or branches in the cross-section.

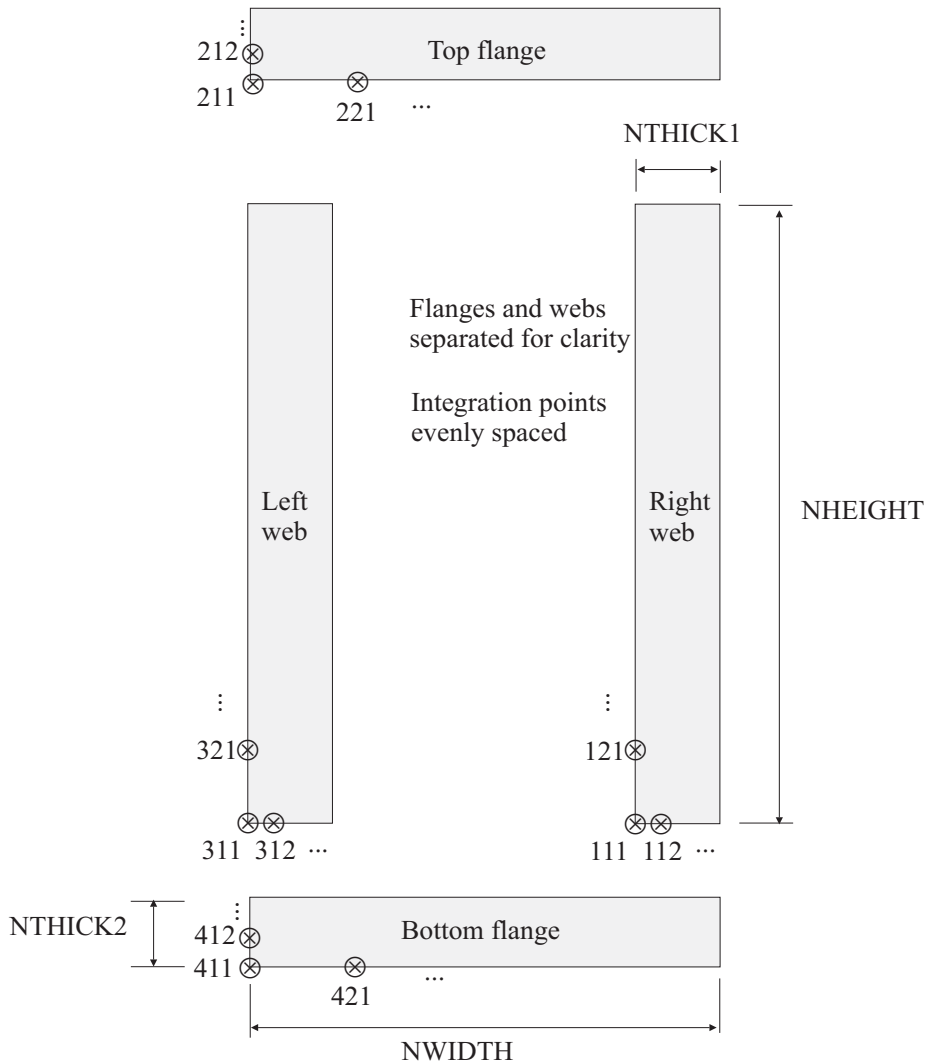


Figure 2.4-22: Numerical integration in the box cross-section

2.4.6.5 I cross-section

Figure 2.4-23 shows the I cross-section and its dimensions. ADINA evaluates the section properties from the given dimensions.

- The I cross-section can be used in both linear elastic and large

displacement elastic analysis. In addition, the I cross-section can be used in MNO and large displacement elastic-plastic analysis.

Table 2.4-5 summarizes the choices available for the I cross-section.

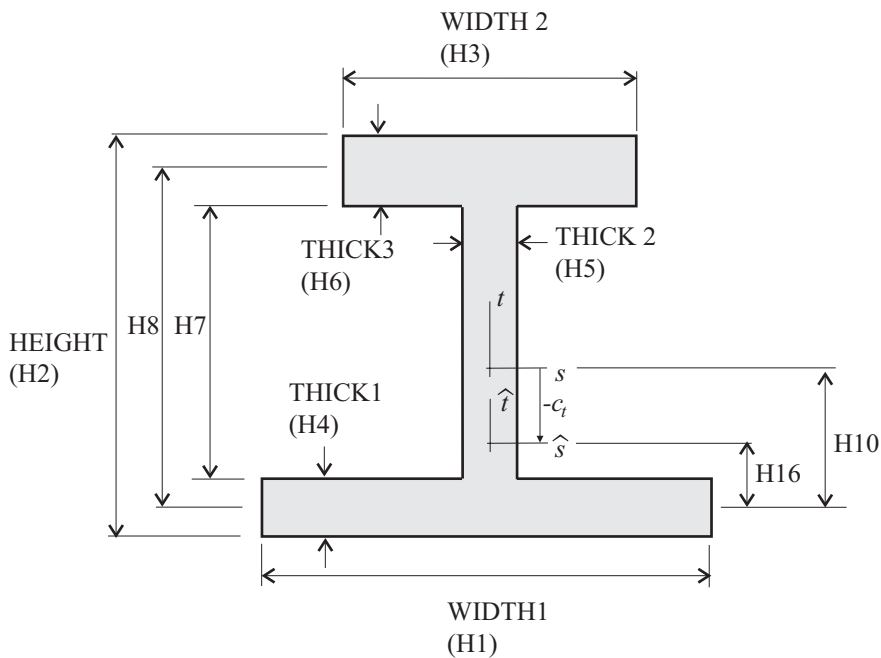


Figure 2.4-23: I cross-section

Table 2.4-5: Options for the I cross-section

	Standard beam, V89 assumptions ¹	Standard beam	Warping beam
Shear center	At neutral axis, shear center effects are ignored	At shear center	At shear center
Formula for calculation of torsional rigidity (elastic material)	F1, F2, FW, default F1	F1, F2, FW, default FW	FW
Warping constant I_{ω} included	No	Yes	Yes
Wagner effect allowed under large displacement conditions	No	Yes, by default	Yes, by default
Plastic material model allowed	No	Yes	Yes

1) CROSS-SECTION I ... ASSUMPTIONS=V89

The table shows that either the standard beam or the warping beam can be used in conjunction with the I cross-section. However, it is not recommended that the standard beam be used with the I cross-section. In order to use the standard beam with the I cross-section, set CROSS-SECTION I ... STANDARD=YES (default is NO).

- When the standard beam is used, there are two choices:

Standard beam, ASSUMPTIONS=V89:

- Offsets to the shear center are not included. Therefore transverse forces applied at the nodes are assumed to be acting through the beam's shear center and hence cause no twisting.
- Warping effects ignored except for the torsional rigidity; mass

matrix effects due to the warping constant I_ω are ignored.

- ▶ Torsional rigidity J calculated by formula F1 by default.
- ▶ Wagner effect not included in large displacement analysis.

Standard beam, ASSUMPTIONS=CURRENT:

- ▶ Offsets to the shear center are included.
- ▶ Warping effects included; the torsional rigidity includes warping effects and mass matrix effects due to the warping constant I_ω are included.
- ▶ Torsional rigidity J calculated by formula FW by default.
- ▶ Wagner effect included by default in large displacement analysis.

General comments

- The calculations of I_ω and c_t are made to be consistent with the calculations done in the plastic material (in which all integrals are evaluated numerically from the warping function). Hence the values of I_ω , c_t might not be the same as the handbook values of these quantities, particularly if the walls of the cross-section are thick.
- The Wagner effect in large displacements is controlled with CROSS-SECTION I ... WAGNER={DEFAULT, YES, NO}. DEFAULT=YES unless the standard beam element is used and ASSUMPTIONS=V89

Comments for elastic analysis

- The torsional rigidity is controlled with CROSS-SECTION I ... TORFORMULA={DEFAULT, F1, F2, FW}.

TORFORMULA=F1 (textbook formula):

$$J = \frac{1}{3} \left(H^4 H_1 + H^6 H_3 + H^5 H_7 \right)$$

TORFORMULA=F2 (alternate textbook formula):

$$J = \frac{1}{3} \left(H^4 H_1 + H^6 H_3 + H^5 H_8 \right)$$

TORFORMULA=FW (formula consistent with assumed warping function):

$$J = \frac{1}{3} (H_4^3 H_1 + H_6^3 H_3 + H_5^3 H_7) \quad (\text{same formula as F1})$$

TORFORMULA=DEFAULT: Use formula FW unless the standard beam is used with ASSUMPTIONS=V89, use formula F1 if the standard beam is used with ASSUMPTIONS=V89.

These formulas assume that the walls of the I section are thin, and then these formulas are numerically almost equal to each other.

When the warping beam is used, formula FW is always used.

J can be multiplied by the user-specified parameter TORFAC.

- Parameters SC, TC are used to adjust the bending moment inertias I_{ss} , I_{tt} of the cross-section, as in the rectangular cross-section.
- Parameters SSHEARF, TSHEARF can be used to calculate the effective shear cross-section areas in the s and t directions, as in the rectangular cross-section.

Comments for elastic-plastic analysis

- The locations and the labeling of the integration points are given in Fig. 2.4-24. For each direction in the cross-section, Newton-Cotes integration is used with 1, 3, 5, 7 integration points in that direction (default 3). The default integration order is sufficient to exactly integrate all quantities in elastic analysis (not considering the Wagner effect). However, the integration order can be increased in order to capture the spread of plasticity throughout the cross-section.

Each integration point is labeled with a four digit number, the first digit giving the r integration point and the last three digits giving the location within the cross-section. For example, integration point 2312 denotes integration point 2 in the r direction and integration point 312 in the cross-section. This integration point is in the top flange of the cross-section.

It can happen that there are two integration points at the same physical location, yet the stresses are different at these points. This happens because the warping function is not continuous at sharp corners or branches in the cross-section.

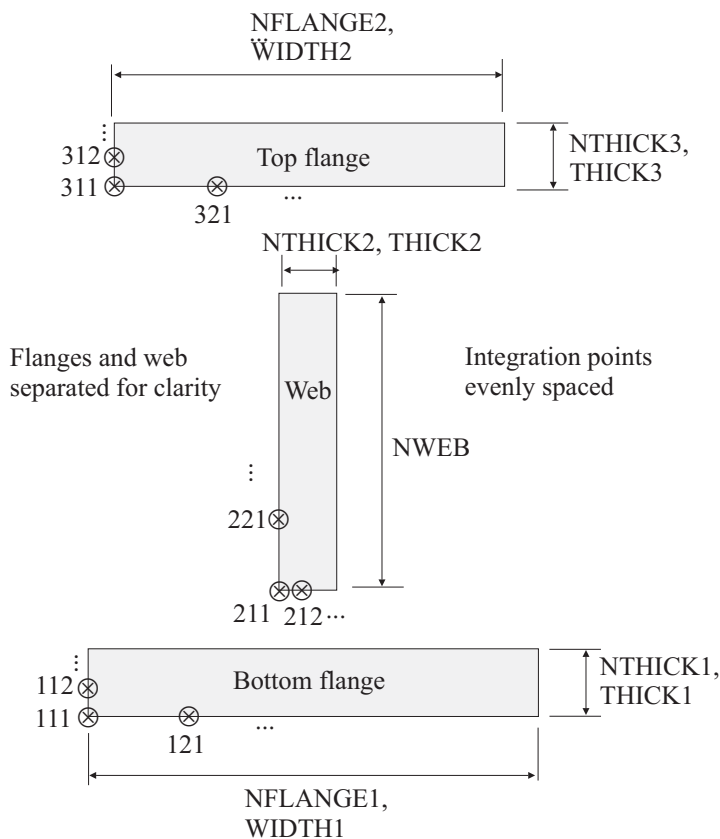


Figure 2.4-24: Numerical integration in the I cross-section

2.4.6.6 U cross-section

Figure 2.4-25 shows the U cross-section and its dimensions. ADINA evaluates the section properties from the given dimensions.

- The U cross-section can be used in both linear elastic and large displacement elastic analysis. In addition, the U cross-section can be used in MNO and large displacement elastic-plastic analysis.

Table 2.4-6 summarizes the choices available for the U cross-section.

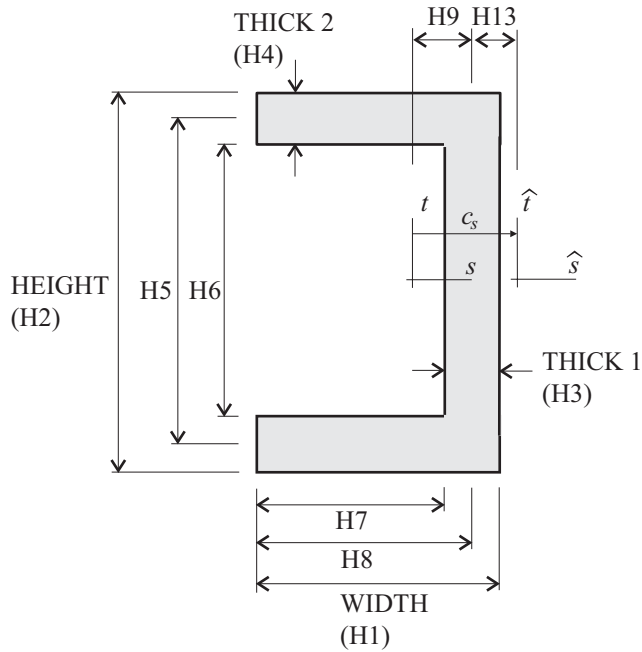


Figure 2.4-25: U cross-section

Table 2.4-6: Options for the U cross-section

	Standard beam, V89 assumptions ¹	Standard beam	Warping beam
Shear center	At neutral axis, shear center effects are ignored	At shear center	At shear center
Formula for calculation of torsional rigidity (elastic material)	F1, F2, FW, default F1	F1, F2, FW, default FW	FW
Warping constant I_{ω} included	No	Yes	Yes
Wagner effect allowed under large displacement conditions	No	Yes, by default	Yes, by default
Plastic material model allowed	No	Yes	Yes

1) CROSS-SECTION U ... ASSUMPTIONS=V89

The table shows that either the standard beam or the warping beam can be used in conjunction with the U cross-section. However, it is not recommended that the standard beam be used with the U cross-section. In order to use the standard beam with the U cross-section, set CROSS-SECTION U ... STANDARD=YES (default is NO).

- When the standard beam is used, there are two choices:

Standard beam, ASSUMPTIONS=V89:

- Offsets to the shear center are not included. Therefore transverse forces applied at the nodes are assumed to be acting through the beam's shear center and hence cause no twisting.
- Warping effects ignored except for the torsional rigidity; mass

matrix effects due to the warping constant I_ω are ignored.

- ▶ Torsional rigidity J calculated by formula F1 by default.
- ▶ Wagner effect not included in large displacement analysis.

Standard beam, ASSUMPTIONS=CURRENT:

- ▶ Offsets to the shear center are included.
- ▶ Warping effects included; the torsional rigidity includes warping effects and mass matrix effects due to the warping constant I_ω are included.
- ▶ Torsional rigidity J calculated by formula FW by default.
- ▶ Wagner effect included by default in large displacement analysis.

General comments

- The calculations of I_ω and c_s are made to be consistent with the calculations done in the plastic material (in which all integrals are evaluated numerically from the warping function). Hence the values of I_ω, c_s used in the elastic material might not be the same as the handbook values of these quantities, particularly if the walls of the cross-section are thick.
- The Wagner effect in large displacements is controlled with CROSS-SECTION U ... WAGNER={DEFAULT, YES, NO}. DEFAULT=YES unless the standard beam element is used and ASSUMPTIONS=V89.

Comments for elastic analysis

- The torsional rigidity is controlled with CROSS-SECTION U ... TORFORMULA={DEFAULT, F1, F2, FW}.

TORFORMULA=F1 (textbook formula):

$$J = \frac{1}{3} (2 H^4 H_1 + H^3 H_6)$$

TORFORMULA=F2 (alternate textbook formula):

$$J = \frac{1}{3} (2 H^4 H_8 + H^3 H_5)$$

TORFORMULA=FW (formula consistent with assumed warping function):

$$J = \frac{1}{3} (2 H^4 t + H^3 H_2)$$

TORFORMULA=DEFAULT: Use formula FW unless the standard beam is used with ASSUMPTIONS=V89, use formula F1 if the standard beam is used with ASSUMPTIONS=V89.

These formulas assume that the walls of the U section are thin, and then these formulas are numerically almost equal to each other.

When the warping beam is used, formula FW is always used.

J can be multiplied by the user-specified parameter TORFAC.

- Parameters SC, TC are used to adjust the bending moment inertias I_{ss} , I_{tt} of the cross-section, as in the rectangular cross-section.
- Parameters SSHEARF, TSHEARF can be used to calculate the effective shear cross-section areas in the s and t directions, as in the rectangular cross-section.

Comments for elastic-plastic analysis

- The locations and the labeling of the integration points are given in Fig. 2.4-26. For each direction in the cross-section, Newton-Cotes integration is used with 1, 3, 5, 7 integration points in that direction (default 3). The default integration order is sufficient to exactly integrate all quantities in elastic analysis (not considering the Wagner effect). However, the integration order can be increased in order to capture the spread of plasticity throughout the cross-section.

Each integration point is labeled with a four digit number, the first digit giving the r integration point and the last three digits giving the location within the cross-section. For example, integration point 2312 denotes integration point 2 in the r direction and integration point 312 in the cross-section. This integration point is in the top flange of the cross-section.

It can happen that there are two integration points at the same physical location, yet the stresses are different at these points. This happens because the warping function is not continuous at sharp

corners or branches in the cross-section.

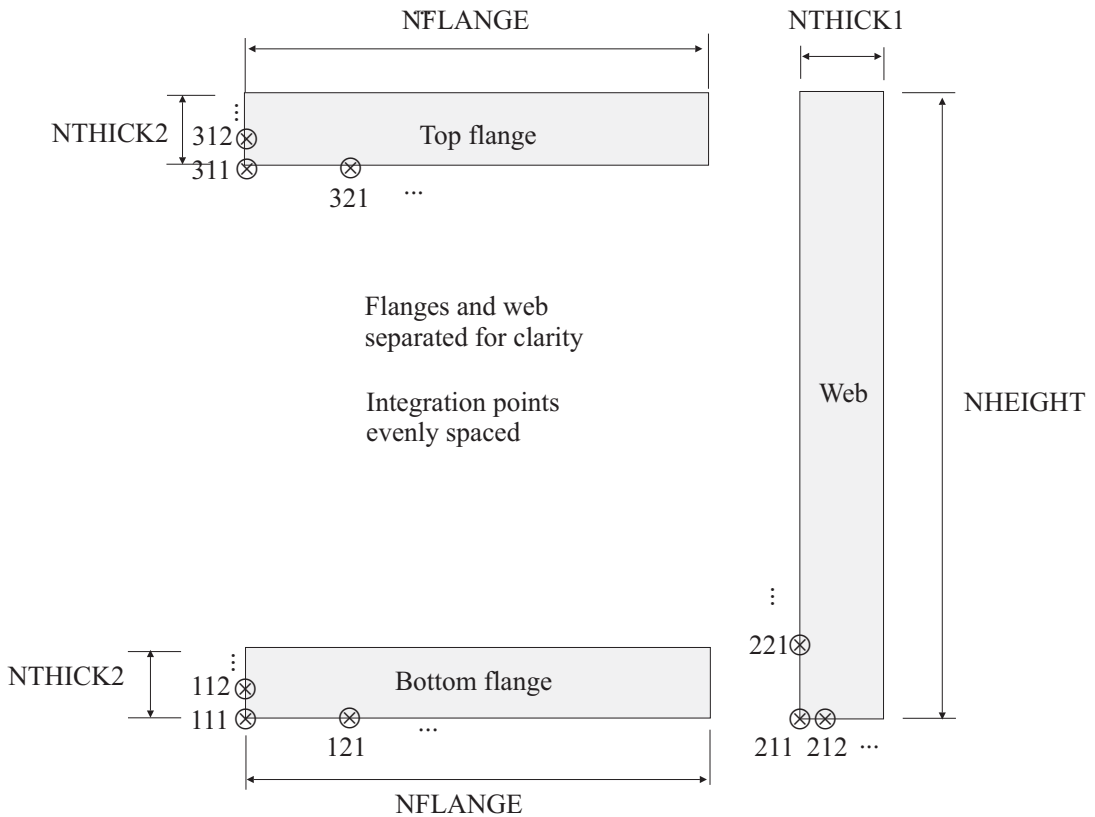


Figure 2.4-26: Numerical integration in the U cross-section

2.4.6.7 L cross-section

Figure 2.4-27 shows the L cross-section and its dimensions. ADINA evaluates the section properties from the given dimensions.

- The L cross-section can be used in both linear elastic and large displacement elastic analysis. In addition, the L cross-section can be used in MNO and large displacement elastic-plastic analysis.

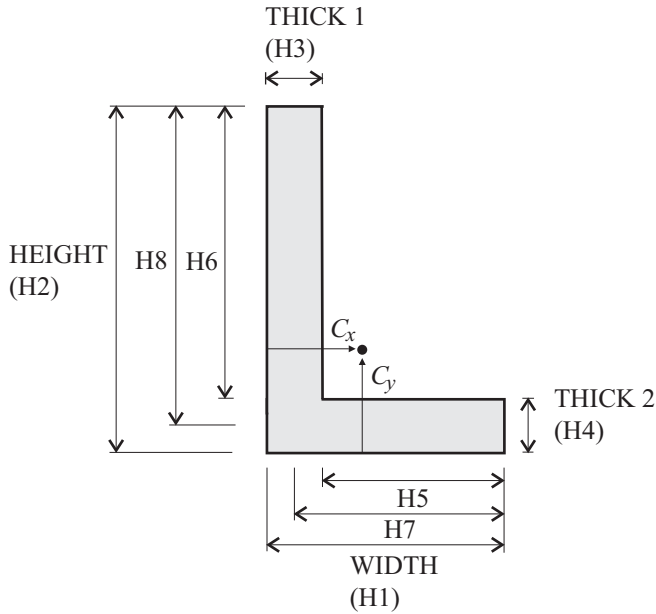
Table 2.4-7 summarizes the choices available for the L cross-section.

Table 2.4-7: Options for the L cross-section

	Standard beam, V89 assumptions ¹	Standard beam	Warping beam
Orientation of cross-section	Principal directions aligned with s-t axes	Horizontal leg aligned with s axis	Horizontal leg aligned with s axis
Shear center	At neutral axis, shear center effects are ignored	At shear center	At shear center
Formula for calculation of torsional rigidity (elastic material)	F1, F2, FW, default F1	F1, F2, FW, default FW	FW
Warping constant I_{ω} included	No	Yes	Yes
Wagner effect allowed under large displacement conditions	No	Yes, by default	Yes, by default
Plastic material model allowed	No	Yes	Yes

1) CROSS-SECTION L ... ASSUMPTIONS=V89

The table shows that either the standard beam or the warping beam can be used in conjunction with the L cross-section. However, it is not recommended that the standard beam be used with the L cross-section. In order to use the standard beam with the L cross-section, set CROSS-SECTION L ... STANDARD=YES (default is NO).



a) Section dimensions

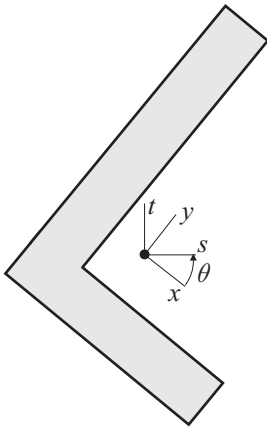
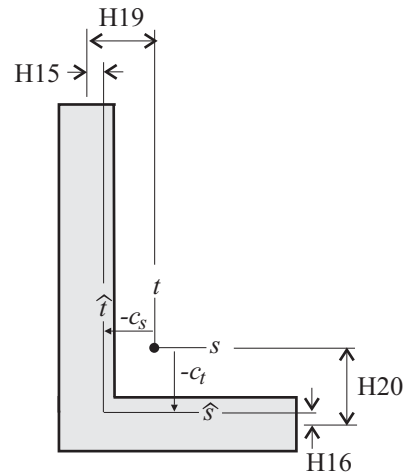
a) Principal axes aligned with s, t axesc) Legs aligned with s, t axes

Figure 2.4-27: L cross-section

- When the standard beam is used, there are two choices:

Standard beam, ASSUMPTIONS=V89:

- ▶ The principal axes are aligned with the s - t axes.
- ▶ Offsets to the shear center are not included. Therefore transverse forces applied at the nodes are assumed to be acting through the beam's shear center and hence cause no twisting.
- ▶ Warping effects ignored except for the torsional rigidity; mass matrix effects due to the warping constant I_ω are ignored.
- ▶ Torsional rigidity J calculated by formula F1 by default.
- ▶ Wagner effect not included in large displacement analysis.

Standard beam, ASSUMPTIONS=CURRENT:

- ▶ The horizontal leg is aligned with the s axis; the vertical leg is aligned with the t axis.
- ▶ Offsets to the shear center are included.
- ▶ Warping effects included; the torsional rigidity includes warping effects and mass matrix effects due to the warping constant I_ω are included.
- ▶ Torsional rigidity J calculated by formula FW by default
- ▶ Wagner effect included by default in large displacement analysis.

General comments

- The calculations of I_ω and c_s, c_t are made to be consistent with the calculations done in the plastic material (in which all integrals are evaluated numerically from the warping function). Hence the values of I_ω, c_s, c_t used in the elastic material might not be the same as the handbook values of these quantities, particularly if the walls of the cross-section are thick.
- The Wagner effect in large displacements is controlled with CROSS-SECTION L ... WAGNER={DEFAULT, YES, NO}. DEFAULT=YES unless the standard beam element is used and ASSUMPTIONS=V89.

Comments for elastic analysis

- The torsional rigidity is controlled with CROSS-SECTION L ...
TORFORMULA={DEFAULT, F1, F2, FW}.

TORFORMULA=F1 (textbook formula):

$$J = \frac{1}{3} (H_3^3 H_2 + H_4^3 H_5)$$

TORFORMULA=F2 (alternate textbook formula):

$$J = \frac{1}{3} (H_3^3 H_8 + H_4^3 H_7)$$

TORFORMULA=FW (formula consistent with assumed warping function):

$$J = \frac{1}{3} (H_3^3 H_2 + H_4^3 H_5) \text{ (same formula as F1)}$$

TORFORMULA=DEFAULT: Use formula FW unless the standard beam is used with ASSUMPTIONS=V89, use formula F1 if the standard beam is used with ASSUMPTIONS=V89.

These formulas assume that the walls of the L section are thin, and then these formulas are numerically almost equal to each other.

When the warping beam is used, formula FW is always used.

J can be multiplied by the user-specified parameter TORFAC.

- When the principal axes are aligned with the s and t axes, the maximum and minimum principal bending moments of inertia (about the s and t axes) are obtained as follows:

$$I_{ss} = \frac{I_{xx} + I_{yy}}{2} + \sqrt{\left(\frac{I_{xx} - I_{yy}}{2}\right)^2 + I_{xy}^2}$$

$$I_{tt} = \frac{I_{xx} + I_{yy}}{2} - \sqrt{\left(\frac{I_{xx} - I_{yy}}{2}\right)^2 + I_{xy}^2}$$

where the moments and products of inertia with respect to the local axes x and y are:

$$\begin{aligned}
I_{xx} &= \frac{H4 H1^3}{12} + H1 H4 \left(\frac{H1}{2} - C_x \right)^2 + \frac{H6 H3^3}{12} + H6 H3 \left(\frac{H3}{2} - C_x \right)^2 \\
I_{yy} &= \frac{H1 H4^3}{12} + H1 H4 \left(\frac{H4}{2} - C_y \right)^2 + \frac{H3 H6^3}{12} + H5 H3 \left(H4 + \frac{H6}{2} - C_y \right)^2 \\
I_{xy} &= H1 H4 \left(\frac{H1}{2} - C_x \right) \left(\frac{H4}{2} - C_y \right) + H6 H3 \left(\frac{H3}{2} - C_x \right) \left(H4 + \frac{H6}{2} - C_y \right)
\end{aligned}$$

and where (C_x, C_y) are the centroid coordinates, measured from the bottom of the section. In this case, the angle between the s principal axis and the horizontal is determined to maximize the following expression: $I_{ss} = I_{xx} \cos^2 \theta + I_{yy} \sin^2 \theta - I_{xy} \sin 2\theta$. The result is

$$\theta = \frac{1}{2} \left(\text{atan2} \left(-I_{xy}, -\frac{1}{2} (I_{yy} - I_{xx}) \right) \right)$$

This angle is shown in Fig 2.4-27.

- When the legs are aligned with the s and t axes, then $\theta = 0$ and

$$I_{ss} = I_{xx}, \quad I_{tt} = I_{yy}, \quad I_{st} = I_{xy}$$

- Parameters SC, TC are used to adjust the bending moment inertias I_{ss} , I_{tt} of the cross-section, as in the rectangular cross-section. When the horizontal leg is aligned with the s axis, I_{st} is also adjusted.
- Parameters SSHEARF, TSHEARF can be used to calculate the effective shear cross-section areas in the s and t directions, as in the rectangular cross-section.

Comments for elastic-plastic analysis

- The locations and the labeling of the integration points are given in Fig. 2.4-28. For each direction in the cross-section, Newton-Cotes integration is used with 1, 3, 5, 7 integration points in that direction (default 3). The default integration order is sufficient to exactly integrate all quantities in elastic analysis (not considering the Wagner effect). However, the integration order can be increased in order to capture the spread of plasticity throughout the cross-section.

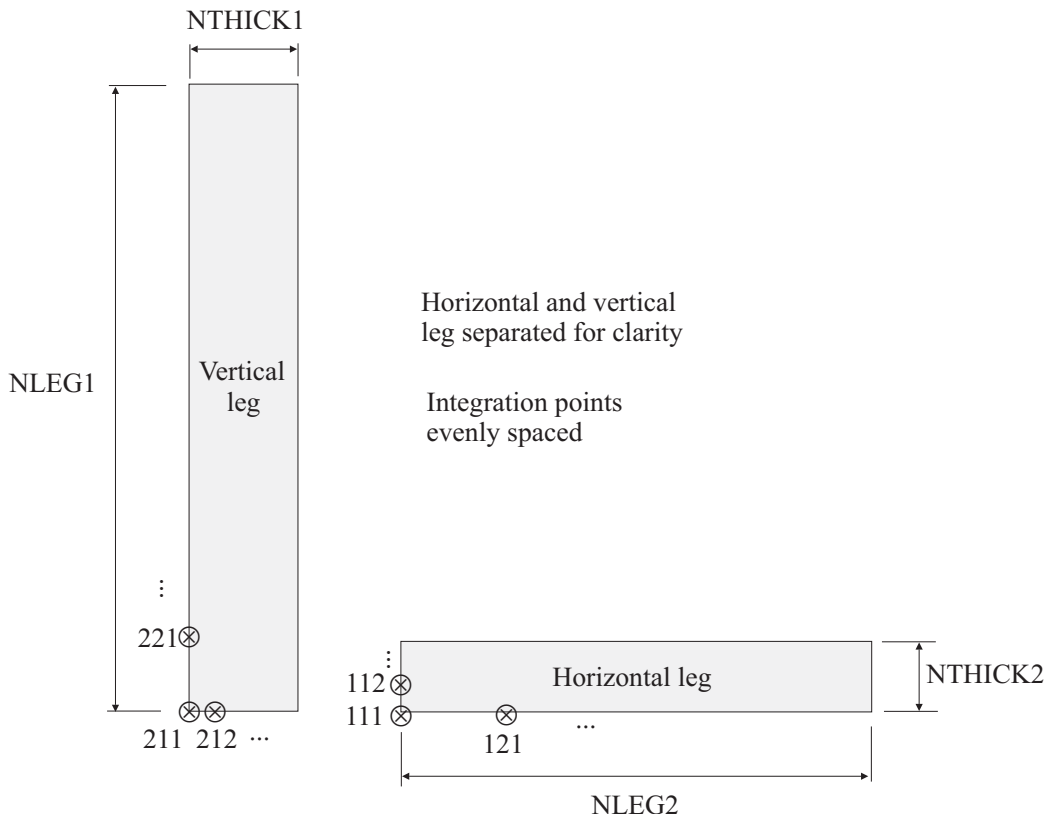


Figure 2.4-28: Numerical integration in the L cross-section

Each integration point is labeled with a four digit number, the first digit giving the r integration point and the last three digits

giving the location within the cross-section. For example, integration point 2212 denotes integration point 2 in the r direction and integration point 212 in the cross-section. This integration point is in the vertical leg of the cross-section.

It can happen that there are two integration points at the same physical location, yet the stresses are different at these points. This happens because the warping function is not continuous at sharp corners or branches in the cross-section.

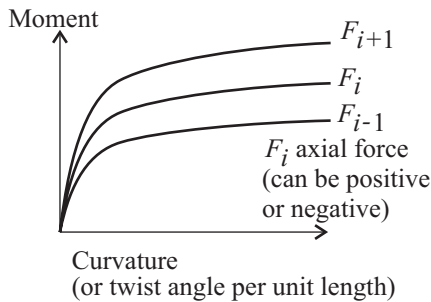
2.4.7 Moment-curvature beam element

- In practical engineering analysis, the data available for the description of the behavior of beam members may be given only in the form of relationships between bending moment and curvature, and between torsional moment and angle of twist. ADINA offers the capability of directly using these data without having to define an "equivalent" stress-strain law and the exact beam cross-sectional shape.

This element is suitable for modeling nonlinear elastic and elasto-plastic beam problems involving arbitrary cross-sections, especially cross-sections that are neither rectangular nor circular.

- The element can be used either with the small displacement formulation (in which case the formulation is materially-nonlinear-only) or with the large displacement formulation (in which case the displacements/rotations can be large). In all cases, the strains are assumed to be small.
- In the moment-curvature input, it is assumed that both the centroid and the shear center of the beam cross-section lie on the r -axis of the element: transverse forces applied to the element cannot generate twisting (as with a shear center offset) and axial forces applied to the element cannot generate bending (as with a centroid offset).
- The flexural and the torsional behavior of the beam element are respectively described by "bending moment vs. curvature" and "torsional moment vs. angle of twist" relationships. These relationships are input in ADINA in the form of multilinear functions (see Figure 2.4-29).

Typical beam data set:



ADINA input:

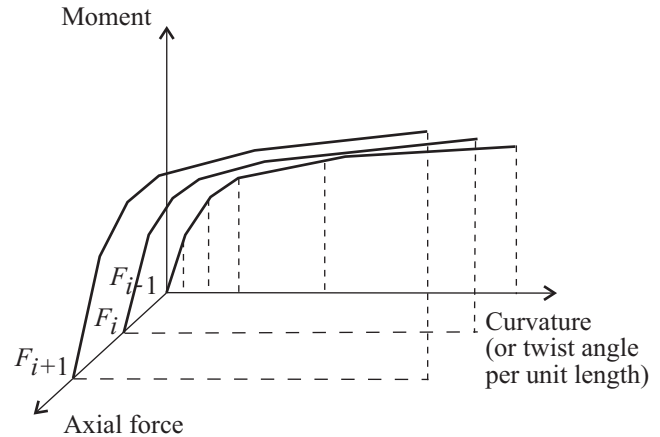


Figure 2.4-29: Input curves for the moment-curvature models (curves are shown only for positive forces, moments and curvatures/twists)

- The "bending moment vs. curvature" and "torsional moment vs. angle of twist" relationships are functions of the axial force. Note that the axial force is positive when the element is in tension and is negative when the element is in compression. The sign conventions used for moments and torsion is that used for local node 2 in Figure 2.4-2.
- The flexural behavior of the beam element is defined by two "bending vs. curvature" relationships, one for each principal plane of inertia. Curvature is defined as $\chi = \frac{1}{\rho}$ where ρ is the radius of curvature. For a linear element (linear elastic material, small displacements/small strains), the relationship between bending moment and curvature is $\frac{M}{\chi} = EI$.
- The input for the torsional behavior of the element is similar to that for bending: A multilinear function is used to express the torsional moment in terms of the angle of twist per unit length.
- Note that the beam element does not include any shear

deformation when the moment-curvature description is used.

- Note also that warping effects must be taken into account in the definition of the "torsional moment vs. twist" input curve if necessary.
- For computation of the beam element stiffness matrix and internal force vector, numerical integration (Newton-Cotes) is only used for integration along the length of the element (r-axis) and no integration is needed over the cross-section (s-axis and t-axis).
- Thermal effects can be included in moment-curvature relations.

Nonlinear elastic model: A nonlinear elastic model can be used (see Figure 2.4-30). In this case, the behavior for negative curvatures (resp. twist angles) may be different than the behavior for positive curvatures (resp. twist angles). The axial force versus axial deformation relationship is always linear elastic.

Note that the last segments of the moment-curvature curve are extrapolated if necessary, in order to calculate the moment when the curvature or twist angle per unit length is out of the input curve range. The end points of the curve do not represent rupture points.

ADINA input for a given axial force:

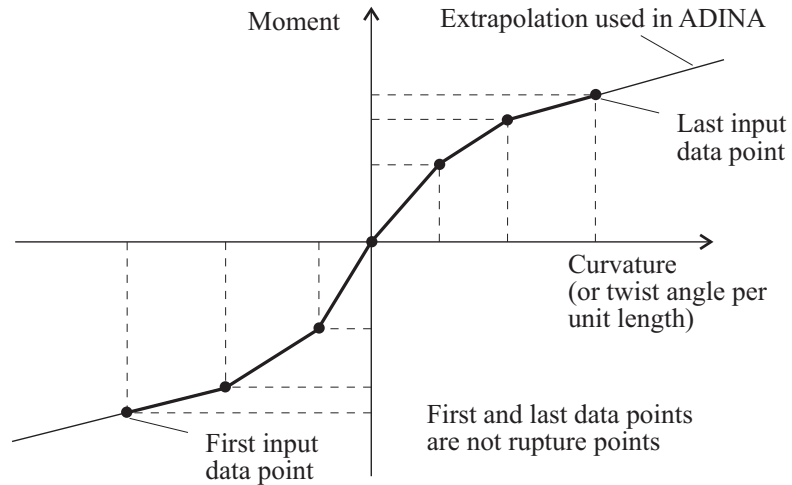
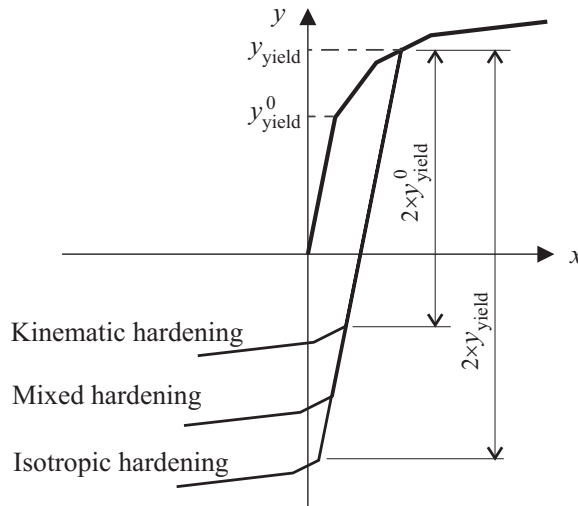


Figure 2.4-30: Nonlinear elastic moment-curvature beam input

Elastic-plastic model: For the analysis of beam members undergoing plastic deformations, ADINA offers bilinear and multilinear plasticity. Hardening can be linear isotropic, linear kinematic or linear mixed, as shown in Figure 2.4-31.



(x, y) : (axial strain, axial force), or
 (curvature, bending moment) for a given axial force, or
 (angle of twist/unit length, torsional moment) for a given
 axial force

Figure 2.4-31: Hardening models for moment-curvature beams

The moment-curvature plasticity model consists of uniaxial plasticity laws respectively applied to the axial strain, each bending curvature, and the twist angle per unit length:

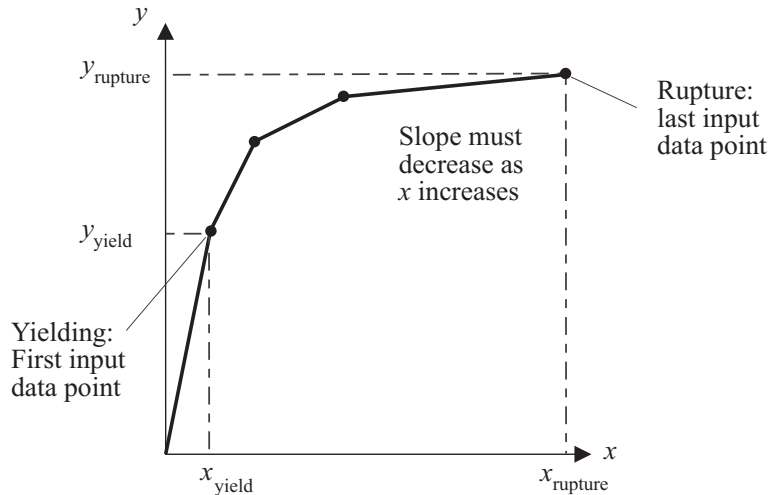
Axial force/axial strain relationship: The relationship can either be symmetric or non-symmetric with respect to the sign of the axial strain.

Bending moment/curvature relationship: The relationship can either be symmetric or non-symmetric with respect to the sign of the curvature. The relationship can depend on the axial force and can be different in axial tension and axial compression.

Torsional moment/twist angle per unit length relationship: The relationship can either be symmetric or non-symmetric with respect to the sign of the twist angle per unit length. The

relationship can depend on the axial force and can be different in axial tension and axial compression.

In the symmetric case (Figure 2.4-32), enter only the positive section of the axial force/axial strain, bending moment/curvature or torsional moment/twist angle curve. The first data point always corresponds to yielding and the last data point always corresponds to rupture.



See Figure 2.4-31 for definition of (x, y)

Figure 2.4-32: Symmetric elasto-plastic moment-curvature beam input

In the non-symmetric case (Figure 2.4-33), enter the entire axial force/axial strain, bending moment/curvature or torsional moment/twist angle curve. The first data point always corresponds to negative rupture and the last data point always corresponds to positive rupture. One data point – the zero point – must be at the origin. The data point prior to the zero point corresponds to the negative yield and the data point after the zero point corresponds to the positive yield. A different number of data points can be used for the positive and negative sections of the curve.

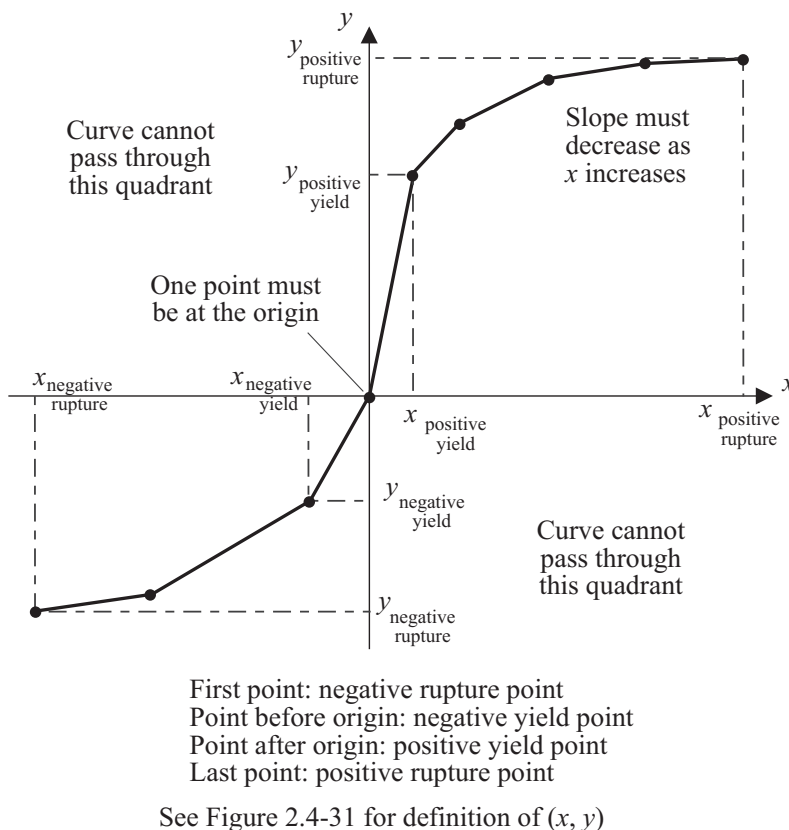


Figure 2.4-33: Non-symmetric elasto-plastic moment-curvature beam input

- To obtain bilinear plasticity, define a multilinear plasticity curve with only two segments.
- Bending moment and torsion relationships can depend on the axial force and this dependence can be different in tension and in compression.
- The input for the bending moment/curvature and torsional moment/twist curves consists of bending moment/curvature curves for different levels of axial force. In the following, we discuss the calculation of the bending moment/curvature curve for a level of axial force that is not input (the calculation of the torsional moment/twist curve is similar).

To obtain the bending moment/curvature curve for a level of axial force not input, interpolation is used. This interpolation is performed, not on the bending moment/curvature curves, but on the bending moment/plastic curvature curves; the bending moment/plastic curvature curves are automatically calculated from the bending moment/curvature curves.

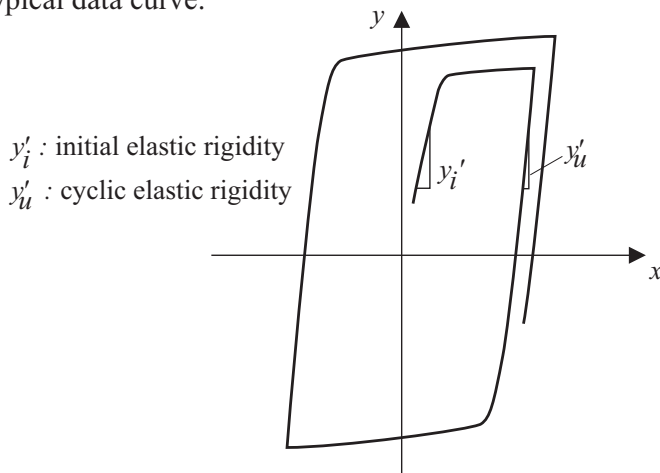
The idea is exactly the same as is used in thermo-plasticity. In thermo-plasticity, the input is a set of stress-strain curves for different temperatures. Interpolation is used to obtain the stress-strain curve for a given temperature. And this interpolation is performed on the stress - plastic strain curves, and not on the stress-strain curves. (See Fig 3.6-3 in Section 3.6.2.)

Because interpolation is performed on the bending moment/plastic curvature curves instead of the bending moment/curvature curves, care should be taken to enter enough bending moment/curvature curves for different axial force levels, especially when the plastic curvature is small for one of the bending moment/curvature curves.

When the response is elastic, the interpolation is performed on the bending moduli.

- The same element section can be plastic with respect to the axial deformation, but still elastic with respect to bending or torsion. The same remark applies for rupture.
- Bending about the two axes is treated independently. Also, there is no interaction between bending and twist. As such, the same element can be plastic in bending about one axis while still elastic with respect to bending about the other axis, or with respect to torsion.
- Rupture depends on the value of the accumulated effective plastic axial strain, accumulated effective plastic curvature or accumulated effective plastic angle of twist per unit length.
- To trigger the rupture calculation of the moment curvature element, the following conditions must be simultaneously met:
 1. number of force-strain points > 2 (asymmetric)
 2. number of twist-moment points > 2 (asymmetric)
 3. number of curvature-moment points > 2 (asymmetric)

Typical data curve:



See Figure 2.4-31 for definition of (x, y)

ADINA result (shown here for the bilinear case):

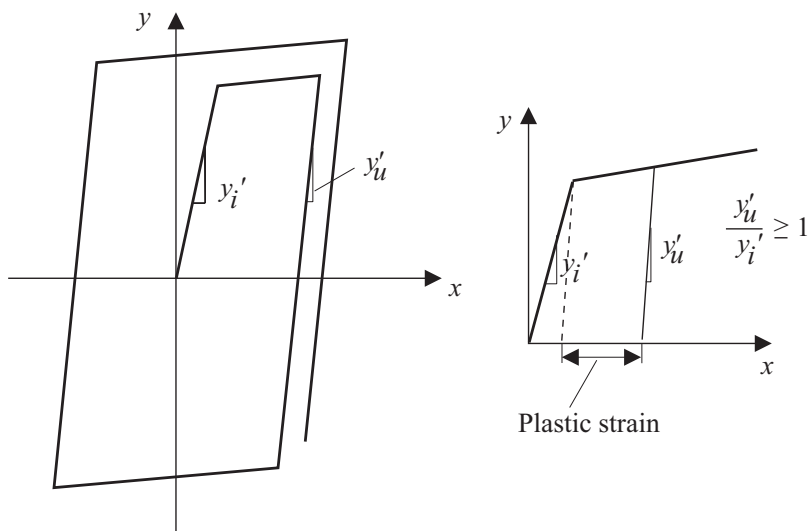


Figure 2.4-34: Typical material curve with a cyclic rigidity different than the initial rigidity

- The elastic-plastic model includes a special option for cyclic behavior, in which the elastic rigidities when unloading first starts differ from the initial rigidities (see Figure 2.4-34). With this option, the elastic rigidity (in bending, torsion or axial deformation)

changes as soon as the yield stress is reached. The ratio of the new cyclic elastic rigidity to the initial elastic rigidity is defined by user input. Once updated, the elastic rigidity remains constant for the rest of the analysis, for loading and unloading.

- In moment-curvature models, the cross-sectional area and the cross-sectional moments of inertia need to be input for the mass matrix computation (in case of mass proportional loading or dynamic analysis). They are used only for mass matrix computation.
- The results can be obtained in the form of element nodal forces and moments, or in the form of stress resultants given at the integration point locations on the element.

Stress resultants: Each element outputs, at the integration point locations on the element r -axis, the following information to the porthole file, based on the material model. This information is accessible in the AUI using the given variable names.

The integration point locations on the element r -axis are shown in Figure 2.4-35. In the AUI, these integration point locations are considered to be section points, not element result points (see Section 13.1.1).

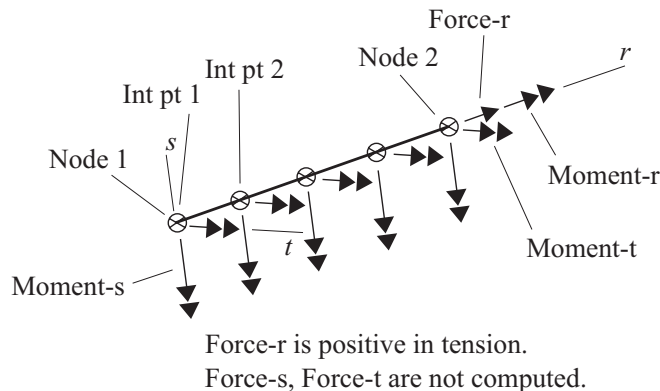


Figure 2.4-35: Stress resultant output option for the moment-curvature models

The results are always given in the element local coordinate system

(r,s,t) . Note that a positive value of the Moment- s at a section point location points in the negative s -direction.

Nonlinear elastic: AXIAL_FORCE, TORSIONAL_MOMENT,
BENDING_MOMENT-S, BENDING_MOMENT-T,
AXIAL_STRAIN, TWIST, CURVATURE-S, CURVATURE-T

Plastic-multilinear: PLASTIC_FLAG_AXIAL,
PLASTIC_FLAG_TORSION, PLASTIC_FLAG_BENDING-S,
PLASTIC_FLAG_BENDING-T, AXIAL_FORCE,
TORSIONAL_MOMENT, BENDING_MOMENT-S,
BENDING_MOMENT-T, AXIAL_STRAIN, TWIST,
CURVATURE-S, CURVATURE-T,
PLASTIC_AXIAL_STRAIN, PLASTIC_TWIST,
PLASTIC_CURVATURE-S, PLASTIC_CURVATURE-T,
ACCUM_PLASTIC_AXIAL_STRAIN,
ACCUM_PLASTIC_TWIST,
ACCUM_PLASTIC_CURVATURE-S,
ACCUM_PLASTIC_CURVATURE-T, YIELD_AXIAL_FORCE,
YIELD_TORSIONAL_MOMENT,
YIELD_BENDING_MOMENT-S,
YIELD_BENDING_MOMENT-T

- Note that in a materially nonlinear analysis, the element nodal moments are in general not equal to the moment stress resultants at the integration points located at the nodes. The reason for this is that in such cases the beam element does not exactly satisfy the element internal equilibrium on the differential level; that is, the element is a "true finite element". The differential equilibrium is, of course, satisfied more accurately as the finite element mesh is refined.
- Transverse forces are not computed at integration points since the ADINA beam element formulation uses the equilibrium equations of the complete element to directly determine the transverse forces at the element end nodes.
- In the case of beam elements with rigid ends, the element stress resultants are computed at the integration point locations corresponding to the flexible part of the element, excluding the rigid ends.

Nodal point forces: Same as for the elastic beam element.

2.4.8 Additional features

End-release option: To model beam internal hinges, a moment and force release option can be used (see Fig. 2.4-36). Twisting moments, axial forces and bimoments (for warping beams) can also be released.

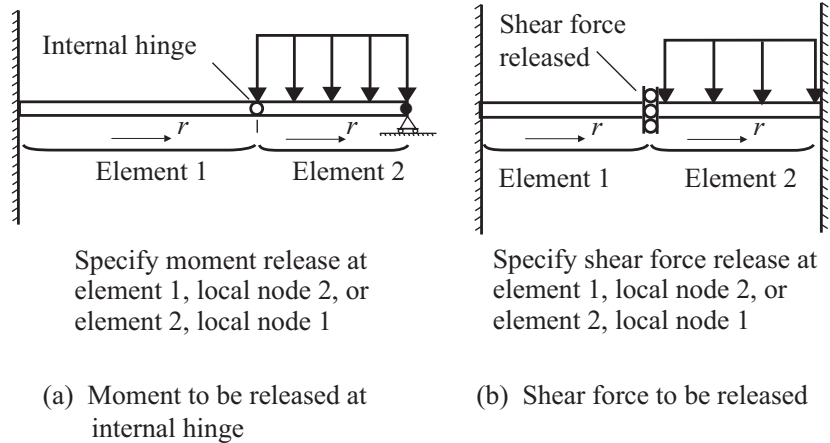


Figure 2.4-36: Use of moment and shear force release options

- A brief description of the theory of end-releases follows:

For static linear elastic analysis without warping degrees of freedom, the stiffness matrix and internal force vector for a beam element can be written as

$$\mathbf{K}\mathbf{u} = \mathbf{F}$$

In this expression, \mathbf{u} contains the displacements and rotations of the beam nodes, in the beam local coordinate system:

$$\mathbf{u}^T = [u_r^1 \quad u_s^1 \quad u_t^1 \quad \theta_r^1 \quad \theta_s^1 \quad \theta_t^1 \quad u_r^2 \quad u_s^2 \quad u_t^2 \quad \theta_r^2 \quad \theta_s^2 \quad \theta_t^2],$$

\mathbf{F} contains the forces and moments of the beam nodes,

$$\mathbf{F}^T = [S_1 \quad S_2 \quad S_3 \quad S_4 \quad S_5 \quad S_6 \quad S_7 \quad S_8 \quad S_9 \quad S_{10} \quad S_{11} \quad S_{12}]$$

and \mathbf{K} is the stiffness matrix. These expressions use the same notations as Figures 2.4-1 and 2.4-2.

Now suppose that one or more of the local displacements or rotations is to be released. “Releasing” means that the corresponding force is set equal to zero. For example, if the axial displacement at local node 2 is to be released, then force S_7 is set to zero.

In order to release the selected local displacements or rotations, the following procedure is used:

- 1) \mathbf{u} is rearranged and partitioned: $\mathbf{u}^T = [\mathbf{u}_A^T \mid \mathbf{u}_B^T]$, so that \mathbf{u}_A contains all of the local displacements / rotations to be retained, and \mathbf{u}_B contains all of the local displacements / rotations to be released. For example, if the axial displacement at local node 2 is to be released, then

$$\mathbf{u}_A^T = [u_r^1 \quad u_s^1 \quad u_t^1 \quad \theta_r^1 \quad \theta_s^1 \quad \theta_t^1 \quad u_s^2 \quad u_t^2 \quad \theta_r^2 \quad \theta_s^2 \quad \theta_t^2]$$

$$\mathbf{u}_B^T = [u_r^2]$$

- 2) \mathbf{F} and \mathbf{K} are similarly rearranged and partitioned. The resulting system of equations is

$$\begin{bmatrix} \mathbf{K}_{AA} & \mathbf{K}_{AB} \\ \mathbf{K}_{AB}^T & \mathbf{K}_{BB} \end{bmatrix} \begin{bmatrix} \mathbf{u}_A \\ \mathbf{u}_B \end{bmatrix} = \begin{bmatrix} \mathbf{F}_A \\ \mathbf{F}_B \end{bmatrix}$$

- 3) The end release condition is now expressed as $\mathbf{F}_B = \mathbf{0}$, and the resulting system of equations becomes

$$\begin{bmatrix} \mathbf{K}_{AA} & \mathbf{K}_{AB} \\ \mathbf{K}_{AB}^T & \mathbf{K}_{BB} \end{bmatrix} \begin{bmatrix} \mathbf{u}_A \\ \mathbf{u}_B \end{bmatrix} = \begin{bmatrix} \mathbf{F}_A \\ \mathbf{0} \end{bmatrix}$$

4) This system of equations is satisfied by choosing

$$\begin{aligned} (\mathbf{K}_{AA} - \mathbf{K}_{AB} \mathbf{K}_{BB}^{-1} \mathbf{K}_{AB}^T) \mathbf{u}_A &= \mathbf{F}_A \\ \mathbf{u}_B &= -\mathbf{K}_{BB}^{-1} \mathbf{K}_{AB}^T \mathbf{u}_A \end{aligned}$$

Therefore the stiffness matrix that is assembled into the global system of equations is $\mathbf{K}_{AA} - \mathbf{K}_{AB} \mathbf{K}_{BB}^{-1} \mathbf{K}_{AB}^T$.

- Any external forces acting onto the released degrees of freedom are ignored.
- When the warping beam is used the warping degrees of freedom can also optionally be released, using the same theory as described above, and the result is that the corresponding bimoments are equal to zero.
- The end release procedure is also implemented for nonlinear analysis, both for materially nonlinear elements and also for geometrically nonlinear elements.
- The end release procedure is available for static, frequency, implicit dynamic, explicit dynamic, and mode superposition analysis.
- End releases only affect the stiffness matrix and force vector, not the mass matrix. Therefore inertial forces and moments (forces and moments due to mass matrix effects) are not released in the above procedure.
- If end releases are used in frequency or dynamic analysis, the elements in which end releases are specified should be very short, in order to minimize the inertial forces and moments.
- End releases are applied to the element local nodes (not to the global nodes). Therefore, to model the hinge shown in Fig. 2.4-

36(a), a moment end release can be applied to local node 2 of element 1, or to local node 1 of element 2 (but not to both local nodes).

The command used to specify end releases is `ENDRELEASE`. For example, the input corresponding to Fig 2.4-36(a) might be

```
ENDRELEASE 1 12
EDATA
ENTRIES EL ENDRERELEASE
1 1
DATAEND
```

This command releases the S_{12} force (t bending moment at local node 2) of element 1.

- In large displacement / large rotation analysis, the global directions corresponding to the released degrees of freedom change as the model deforms. By default, this effect is fully included in the end release calculations. However, this effect can occasionally slow down convergence. For this reason, this effect can be controlled using the options `EGROUP BEAM ENDRERELEASE=(ACCURATE, APPROX1, APPROX2)`.

If `ENDRELEASE=ACCURATE` (the default), the end releases are evaluated in a local coordinate system corresponding to the configuration of the beam in the previous equilibrium iteration. Therefore, at convergence, the end releases are evaluated in (almost) the current configuration of the beam. (At convergence, the difference between the current equilibrium iteration and the previous equilibrium iteration is very small.)

If `ENDRELEASE=APPROX1` or `APPROX2`, the end releases are evaluated in a local coordinate system corresponding to the configuration of the beam in the previous time step. Notice that the local coordinate system does not change directions during the equilibrium iterations. In this case, the released forces/moments are not exactly zero at convergence, but these released forces/moments should be small, especially if the time step size is small. Compared to `APPROX1`, `APPROX2` includes an additional approximation that reduces accuracy, but improves convergence. Thus `ACCURATE` is more accurate than `APPROX1` and `APPROX1` is more accurate than `APPROX2`; however `APPROX2` converges better than `APPROX1` and `APPROX1` converges better

than ACCURATE.

- In explicit dynamic analysis, the use of end releases will cause the program to run considerably more slowly than if this option were not used.

Rigid-end option: For the modeling of assemblages of braces and struts, it is sometimes of interest that the joints of the structure be considered as infinitely rigid (see Figure 2.4-37). This condition can be modeled using the rigid-end option: with this option, a fraction of each end of the element can be considered as rigid.

End 1 is the end at node 1, end 2 is the end at node 2. It is allowed for end 1 only to be rigid, end 2 only to be rigid, or both ends to be rigid.

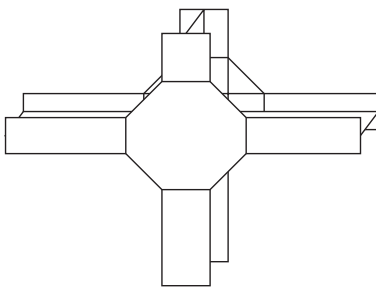
The lengths of the beam rigid ends are specified in the units of the model.

One possible command input for the beam shown in Fig 2.4-38 is

```
EDATA
ENTRIES EL RIGID1 RIGID2
1 0.2 0.4
DATAEND
```

In this example, the length of rigid end 1 is 0.2 units, and the length of rigid end 2 is 0.4 units.

Physical problem:



Finite element model:

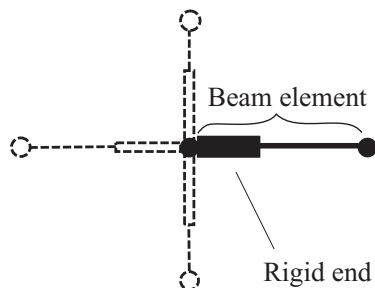


Figure 2.4-37: Modeling of a beam intersection using elements with rigid ends



Figure 2.4-38: Example of beam element rigid ends input

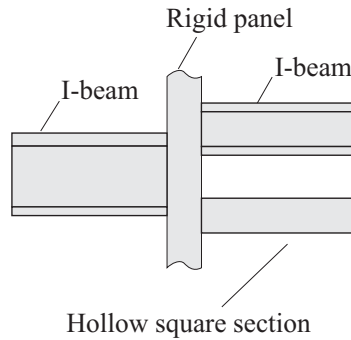
- The assumptions of the rigid-end option are as follows:
 - ▶ In ADINA 8.8 and later, the rigid ends are assumed to be perfectly (infinitely) rigid. In ADINA 8.7 and earlier, the rigid ends are either assumed to be perfectly rigid, or only very stiff (with rigidity calculated by multiplying the rigidity of the element by a large number, such as 10^6).
 - ▶ The element, including its rigid ends, is formulated as one single element. Hence, as for all beam elements, the internal deformations and element internal forces are all referred to a local coordinate system based on the end-nodes, and using the assumptions of small internal relative deformations.
 - ▶ The rigid ends will never undergo plasticity. Hence, although the stress in the rigid ends may be very high (they are not printed by the program), plasticity is not reached. This may be quite unrealistic in a model using the rigid-end capability.
 - ▶ The mass of the rigid ends is taken into account in the mass matrix calculations.
- Rigid ends can be used in all beam analyses.
- When using warping beams with rigid ends, the warping degree of freedom should be fixed at the rigid end.

2.4.9 Beam element modeling hints

- For modeling tapered beams or curved beams, the user needs to divide such members into several elements. For the tapered beams the user needs to divide the beam into several elements and use an appropriate constant cross-section for each of these elements.

- Off-centered beam elements can be created using rigid links (see Fig. 2.4-39 and Section 5.15.2).

Physical problem:



Finite element model:

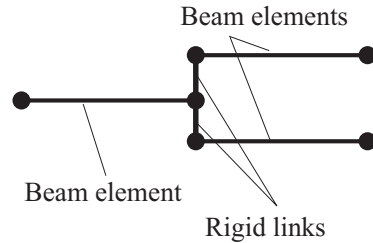


Figure 2.4-39: Use of rigid links for modeling of off-centered beams

- In order to model the bending due to an off-centroidal axial force or transverse load, either apply the resulting moments directly or apply the forces at an offset location using rigid links (Figure 2.4-40).

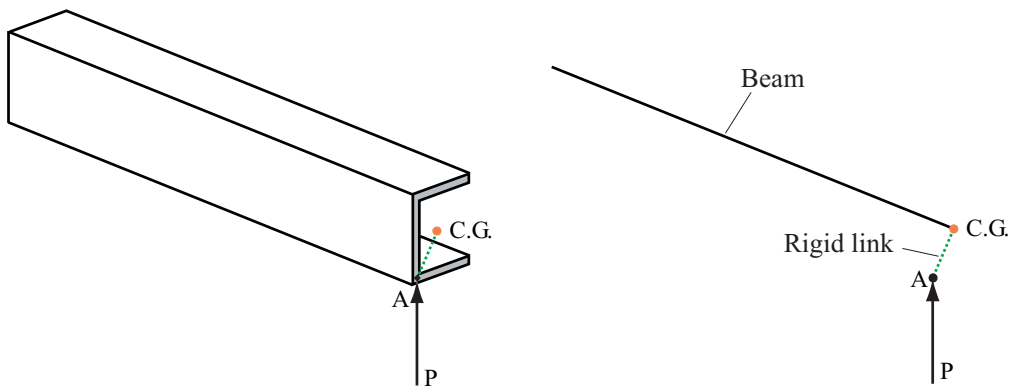


Figure 2.4-40: Using rigid links to transfer loads to the centroid of the beam

- When solving problems involving flexural buckling, it may be necessary to use a refined mesh. This is because the only second-order strain terms included in the formulation are the terms needed

to model the Wagner effect. Second-order strain terms corresponding to flexural buckling effects are not included in the formulation. However, by refining the mesh, it is possible to obtain accurate flexural buckling effects.

- In the moment-curvature beam element, when using nonsymmetric relationships, such as a nonsymmetric axial force - axial strain relationship, it is helpful to apply a very small load in the first step.

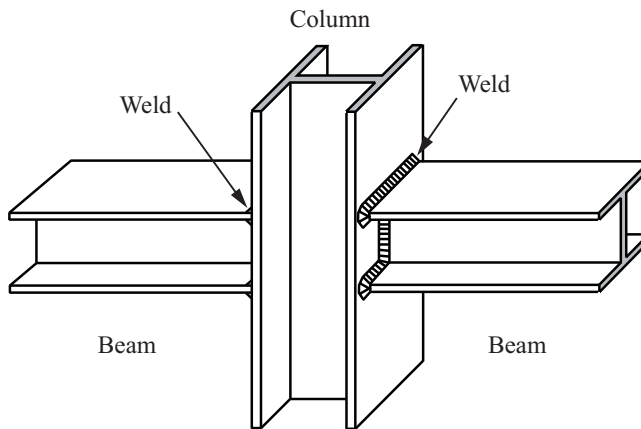
Consider a scenario in which the axial force - axial strain curve has different moduli for tensile and compressive strains. At the beginning of the first step, the strain is zero, and the program uses the tensile modulus in this case. However, if the loading is compressive, then the "wrong" modulus is used, and if the load is large enough, the subsequent iterations do not converge. On the other hand, if the load is very small, then even though the "wrong" modulus is used in the first iteration, the response remains elastic, the correct modulus is used thereafter and the iterations converge. After the first step, the correct modulus is used, so the load increment can be large after the first step.

2.4.9.1 Hints concerning the warping degrees of freedom

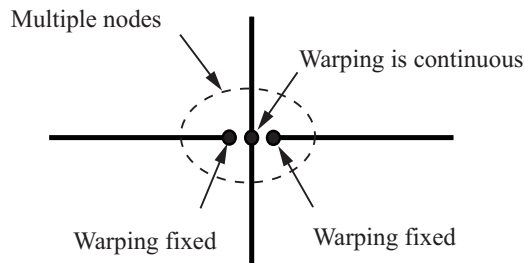
- The effects of fixing the warping can be significant for thin open section beams.
- When defining the fixity for the warping beams, it is important to know what type of physical condition leads to fixed warping in a support and when the warping at a support is free. As an example, when only the web of an I-beam is connected to the column while its flanges are free to move out of their original plane, warping is free. However, if the flanges of the beam are welded to the column, they cannot move out of plane and warping is fixed.
- For a structural connection where several beams and columns are connected at a point it is important to know how to assign the warping degree of freedom. Fig. 2.4-41 shows one such example. As can be seen, the warping degree of freedom is continuous for the columns. However, warping for the beams at the point of connection to column should be set to zero as their flanges are welded to the column and cannot move out of plane. As such, the

user must define multiple nodes at the point of connection, assign the warping degree of freedom for each node separately and establish proper constraint equations between these multiple nodes to ensure that they move and rotate together.

- Also, in modeling curved members using straight warping beams special care should be taken regarding the continuity of the warping degrees of freedom. The angle between any two consecutive elements should be small to ensure the continuity of warping displacements (Fig. 2.4-42).



(a) Actual beam-column connection



(b) Warping beam representation

Figure 2.4-41: Modeling of several warping beams connected at a sharp angle

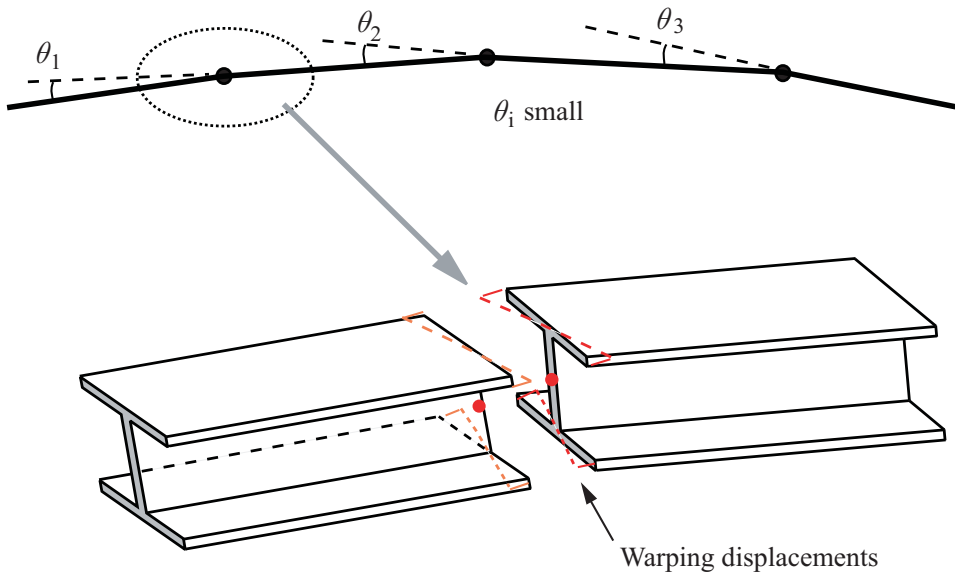


Figure 2.4-42: Discretization of a curved member using straight warping beams

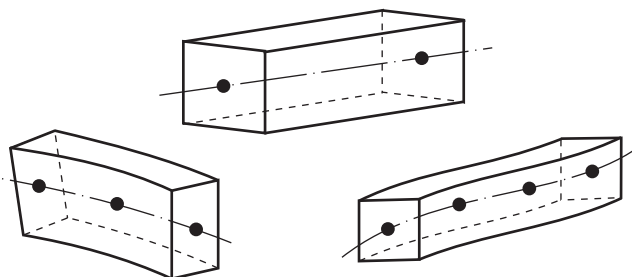
- In general, continuity of the warping deformation at a point where several members meet not only depends on the angle at which these members meet but also on the details of the arrangement of the connector elements (bolts, welds, etc.)
- Warping degrees of freedom cannot be prescribed. In other words, warping degrees of freedom can only be fixed or free.
- Warping degrees of freedom cannot be used in constraint equations or rigid links.
- Bimoments cannot be specified as nodal loads.

2.5 Iso-beam elements; axisymmetric shell elements

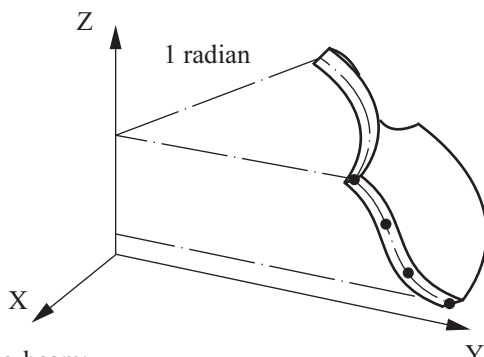
2.5.1 General considerations

- The iso-beam elements (which include the axisymmetric shell elements) can be employed in the following forms, see Fig. 2.5-1:
plane stress 2-D beam with three degrees of freedom per node,
plane strain 2-D beam with three degrees of freedom per node,
axisymmetric shell with three degrees of freedom per node,
general 3-D beam with six degrees of freedom per node.
- The plane stress 2-D beam element is identical to the 3-D beam element but constrained to act only in the YZ plane. Hence, all motion of the 2-D plane stress beam element must occur in the YZ plane (see Fig. 2.5-2).
- The difference between the plane strain and the plane stress 2-D beam elements is that for the plane strain element, it is assumed that the out-of-plane strain ϵ_{xx} is equal to zero whereas the out-of-plane stress σ_{xx} is equal to zero for the plane stress element.
- Note that it can be significantly more effective to use the 2-D beam option instead of the general 3-D beam option, since then the numerical integration is only performed in two dimensions and the number of degrees of freedom is also reduced.

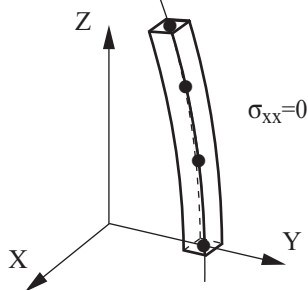
Examples of general 3-D iso-beam elements:



Axisymmetric shell:



Plane stress iso-beam:



Plane strain iso-beam:

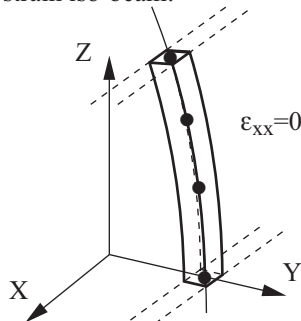
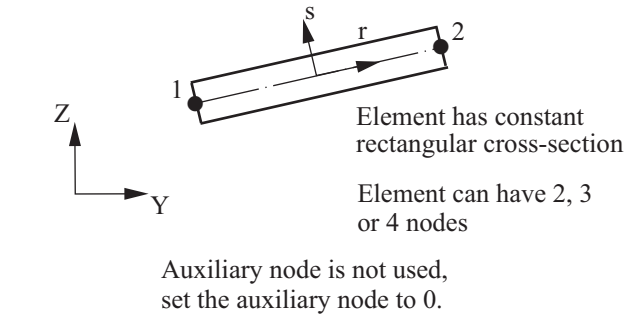
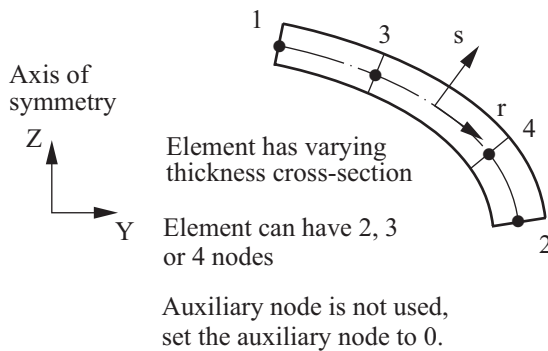


Figure 2.5-1: 2-D iso-beam, axisymmetric shell, and 3-D iso-beam elements



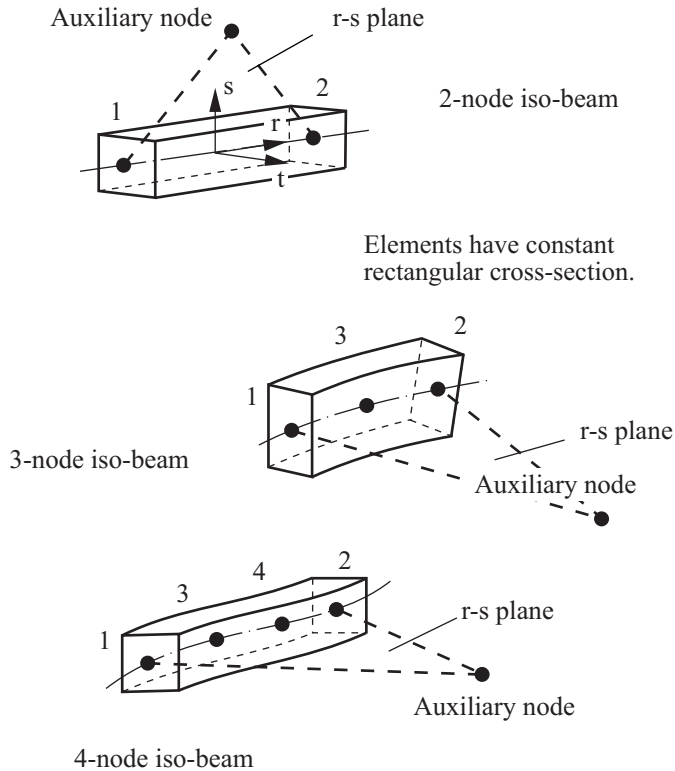
a) 2-node plane stress or plane strain beam element



b) 4-node axisymmetric shell element

Figure 2.5-2: Local node numbering; natural coordinate system

- The axisymmetric shell element formulation is an extension of the 2-D beam element formulation in that the axisymmetric hoop stress/strain components are included in the model. The axisymmetric shell element must be defined in the YZ plane, and lie in the +Y half plane.
- The cross-sectional areas of each of these elements are assumed to be rectangular. The 2-D and 3-D beam elements can only be assigned a constant area cross-section. The axisymmetric shell element can be assigned a varying thickness.



c) General 3-D isoparametric beam elements.

Figure 2.5-2: (continued)

- The elements can be employed with 2, 3 or 4 nodes. The 3 and 4-node elements can be curved, but it should be noted that the element nodes must initially lie in one plane (which defines the r - s plane).
- To model the torsional stiffness of the 3-D beam element accurately, the warping behavior is represented as described in the following reference:

ref. K.J. Bathe and A. Chaudhary, "On the Displacement Formulation of Torsion of Shafts with Rectangular Cross-Sections", *Int. J. Num. Meth. in Eng.*, Vol. 18, pp. 1565-1568, 1982.

- Note that the formulation directly models shear deformations in an approximate manner; namely, the shear deformations are assumed to be constant over the cross-section (see ref. KJB, Fig. 5.18, p. 398). This corresponds to using shear factors equal to 1.0. The elements can be employed to model thin and thick beams and shells.

- Some applications using the element are published in the following reference.

ref. K.J. Bathe and P.M. Wiener, "On Elastic-Plastic Analysis of I-Beams in Bending and Torsion",
Computers and Structures, Vol. 17, pp. 711-718, 1983.

- The isoparametric beam element is available in ADINA primarily to model
 - ▶ Curved beams
 - ▶ Stiffeners to shells, when the shell element (described in Section 2.7) is used
 - ▶ Beams in large displacements
 - ▶ Axisymmetric shells under axisymmetric loading
- Considering the analysis of thick curved beams, in order to represent the hyperbolic stress distribution through the thickness, the higher order elements (3 or 4-node elements) should be used. This is because the nodal director vectors are constructed from the geometry of the elements.

2.5.2 Numerical integration

ref. KJB
Sections 5.4.1
and 6.5.1

- The element matrices and vectors are formulated using the isoparametric interpolation, and Gauss or Newton-Cotes numerical integration is used to evaluate these matrices in all analyses. For the 2-D beam and axisymmetric shell elements, numerical integration is only performed in the r-s plane, hence these elements are considerably less expensive in terms of computer time than the general 3-D element. The locations and labeling of the integration

points are illustrated in Fig. 2.5-3. INR is the index giving the location of the integration point in the r direction, with $\text{INR}=1$ corresponding to the lowest value of r , and $\text{INR}=(\text{integration point order})$ corresponding to the highest value of r . INS and INT are analogous for the s and t directions.

- The elements are based upon a mixed interpolation of displacements and stresses.

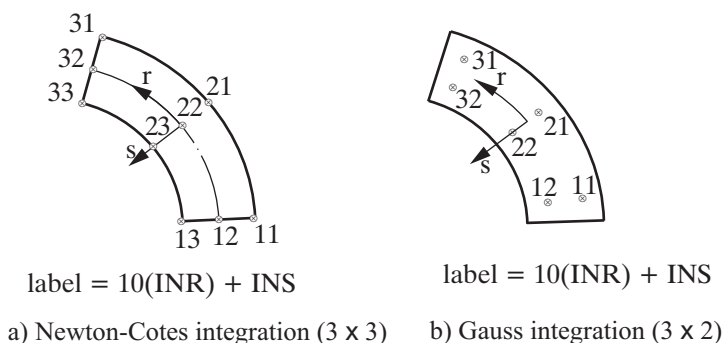


Figure 2.5-3: Examples of integration point numbering for 2-D beam, axisymmetric shell, and 3-D beam elements

- Only the default integration order along the r -direction should be used for the 2-, 3- and 4-node elements, i.e.:

- ▶ 2-node iso-beam: 1-point integration
- ▶ 3-node iso-beam: 2-point integration
- ▶ 4-node iso-beam: 3-point integration

These elements do not contain any spurious zero energy modes, do not lock and are efficient in general nonlinear analysis.

- If, however, the default r -integration order is not used, the results of an analysis can change drastically, for example, if 2-point integration along the r direction is specified for the 2-node element model shown in Fig. 2.5-4, the resulting displacement (Δ) is 0.0126

(i.e., locking is observed).

- For the 3-D beam element, you can choose either 4×4 Gauss or 7×7 Newton-Cotes integration over the beam cross-section (along s and t).

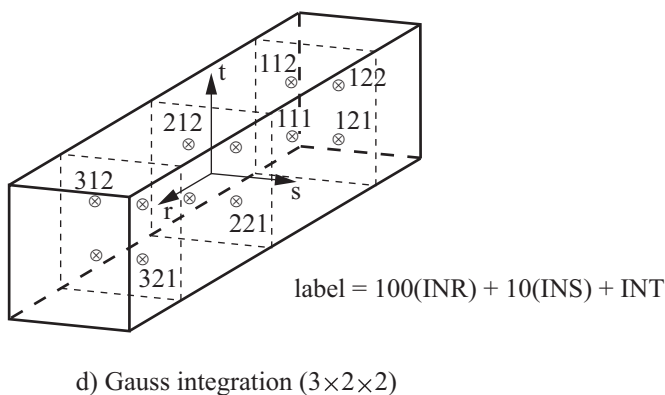
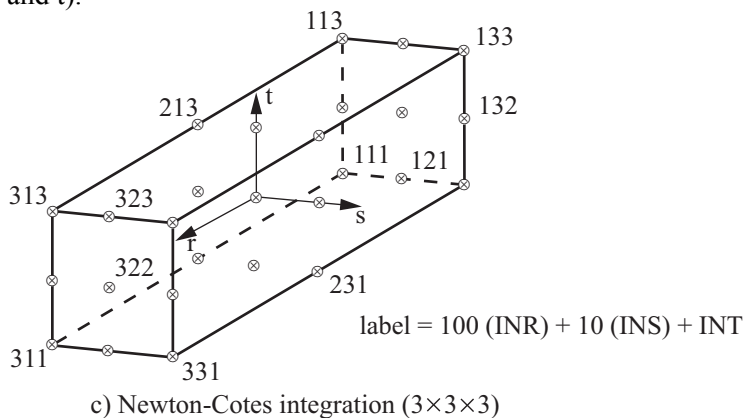


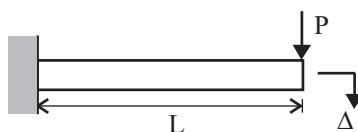
Figure 2.5-3: (continued)

2.5.3 Linear iso-beam elements

ref. KJB
Section 5.4.1

- The formulation of the element is presented in Chapter 5 of ref. KJB.
- It is assumed that the displacements, rotations, and strains are infinitesimally small and the elastic-isotropic material is used.

Physical problem:



$$EI=10^5$$

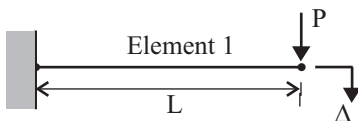
$$L=24$$

$$P=20$$

$$\Delta_{th} = \frac{PL^3}{3EI} = 0.9216$$

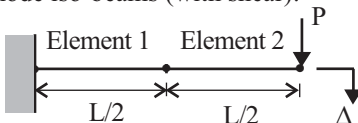
(theoretical solution,
no shear)

2-node Hermitian beam (no shear):



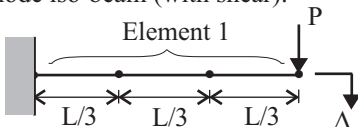
$$\Delta = 0.9216$$

2-node iso-beams (with shear):



$$\Delta = 0.8648$$

4-node iso-beam (with shear):



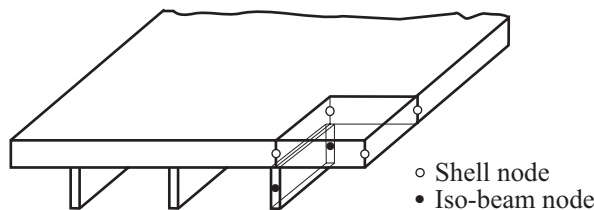
$$\Delta = 0.9224$$

Figure 2.5-4: Results of the analysis of a cantilever

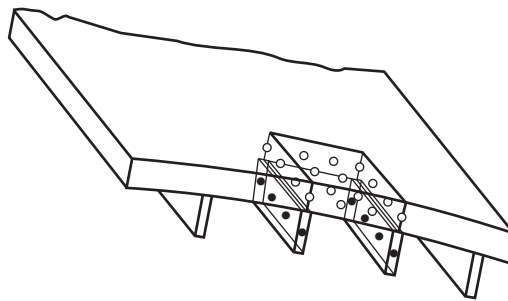
- Since the iso-beam element stiffness matrix is calculated using numerical integration, it is clear that the 2-node Hermitian beam element (see Section 2.4.1) is more effective when straight beam members in linear elastic analysis are considered. Hence, for the linear analysis of many structures (such as frames, buildings, shafts) only the 2-node Hermitian beam element should be used.
- Even when considering the linear analysis of a curved beam member, it is frequently more cost-effective to model this beam as an assemblage of straight 2-node Hermitian beam elements than to use the curved iso-beam element.
- Note that the iso-beam element can only be employed with a rectangular cross-section, whereas the linear 2-node Hermitian

beam element can be used for any cross-section which can be defined by the principal moments of inertia, cross-sectional area and the shear areas input to ADINA.

- In linear analysis, the iso-beam element is primarily useful for modeling stiffeners to shells when the 4-node or 8-node or 16-node shell element (see Section 2.7) is employed. In this case the 2-, 3- or 4-node iso-beam elements, respectively, together with the shell elements can provide an effective finite element discretization of the stiffened shell structure (see Fig. 2.5-5).



a) Stiffened plate



b) Stiffened shell

Note: Select master nodes and use rigid links to tie the shell nodes and the iso-beam nodes together.

Figure 2.5-5: Models of stiffened plate and stiffened shell

- The solution results obtained using the isoparametric beam elements are compared with those obtained using the 2-node

Hermitian beam elements in Fig. 2.5-4. Considering the tip loaded cantilever shown, a 2-node Hermitian beam element representation gives the "exact" analytical value of displacements and section forces/moments (compared with Bernoulli-Euler beam theory).

The 2-node iso-beam element yields approximate results, while the 4-node iso-beam element is very accurate. However, as is typical in finite element analyses, the discrepancy in the solution becomes negligibly small when a sufficiently fine mesh is used.

- The forces/moments at the element nodes are (in linear analysis)

$$\mathbf{F} = \mathbf{K}\mathbf{U}$$

where \mathbf{K} is the element stiffness matrix and \mathbf{U} is the vector of nodal point displacements. Hence, at internal element nodes the forces/moments in \mathbf{F} will be equal to the forces/moments, applied externally or by adjoining elements, and are not the internal section forces/moments.

2.5.4 Nonlinear iso-beam elements

ref. KJB • The formulation of the element is presented in Chapter 6 of ref.
Section 6.5.1 KJB.

- The element can be used with the following material models: **elastic, plastic-bilinear, plastic-multilinear, thermo-isotropic, thermo-plastic, creep, plastic-creep, multilinear-plastic-creep, creep-variable, plastic-creep-variable, multilinear-plastic-creep-variable, shape-memory alloy.**
- The element can be used either with a **small displacement** or a **large displacement** formulation. In the large displacement formulation, large displacements and rotations are allowed. In all cases, only small strains are assumed.

All of the material models in the above list can be used with either formulation. The use of a linear material with the small displacement formulation corresponds to a linear formulation, and the use of a nonlinear material with the small displacement formulation corresponds to a materially-nonlinear-only formulation.

- The iso-beam element can be particularly effective in geometric nonlinear analysis of straight and curved members, because the change in geometry due to the large displacements is modeled accurately.
- ref. KJB
Section 6.6.3
- The element matrices are evaluated using Gauss or Newton-Cotes numerical integration. In elastic-plastic analysis, the stress-strain matrix is modified to include the effects of plasticity. This stress-strain matrix is based on the classical flow theory with the von Mises yield condition and is derived from the three-dimensional stress-strain law with the appropriate stresses and strains set to zero.
 - For the elasto-plastic and the thermo-elastic-plastic and creep material models, the effective-stress-function algorithm of the references given below is used.
- ref. M. Kojić and K.J. Bathe, "The Effective-Stress-Function Algorithm for Thermo-Elasto-Plasticity and Creep", *Int. J. Num. Meth. Engng.*, Vol. 24, No. 8, pp. 509-532, 1987.
- ref. M. Kojić and K.J. Bathe, "Thermo-Elastic-Plastic and Creep Analysis of Shell Structures", *Computers & Structures*, Vol. 26, No 1/2, pp. 135-143, 1987.

2.5.5 Axisymmetric shell element

- The axisymmetric shell element can be thought of as a special case of the iso-beam formulation (in the same way as an axisymmetric element is a special case of the 2-D solid element).
- For the axisymmetric shell element, the hoop (circumferential) stress and strain components are included in the iso-beam formulation.
- The use of this element can be significantly more effective than using axisymmetric 2-D solid elements when the shell structure to be modeled is thin.

- One radian of the axisymmetric shell is modeled when using this element (as with the 2-D axisymmetric solid element).

2.5.6 Element mass matrices

- The iso-beam element can be used with a lumped or a consistent mass matrix, except for explicit dynamic analysis which always uses a lumped mass.
- ref. KJB
Section 5.4.1* • The consistent mass matrix is calculated using the isoparametric formulation with the displacement interpolations given on p. 408 of ref. KJB.

- The lumped mass for degree of freedom i is $M \cdot \left(\frac{\ell_i}{L} \right)$, where M is the total mass, L is the element length and ℓ_i is a fraction of the length associated with node i . The rotational mass is $\frac{1}{3} \cdot M \cdot \frac{\ell_i}{L} \left(\frac{1}{6} (b^2 + d^2) \right)$ where b, d are the width and depth of the cross-section. For a 2-node iso-beam element $\ell_1 = \ell_2 = \frac{L}{2}$, while for a 4-node iso-beam element $\ell_1 = \ell_2 = \frac{L}{6}$ and $\ell_3 = \ell_4 = \frac{L}{3}$ (see Fig. 2.5-6).

- The rotational lumped mass can be multiplied by a user-specified scalar ETA (except in explicit dynamic analysis).

2.5.7 Element output

- You can request that ADINA either print or save stresses or forces.

Stresses: Each element outputs, at its integration points, the following information to the porthole file, based on the material model. This information is accessible in the AUI using the given

variable names.



a) 2-node element



b) 3-node element

Note: M = total mass of element (2-D or 3-D beam elements)



c) 4-node element

Figure 2.5-6: Construction of element lumped translational mass matrix for the beam option of the iso-beam element

General 3-D iso-beam, elastic-isotropic material: STRESS-RR, STRESS-RS, STRESS-RT, STRESS-SS, STRAIN-RR, STRAIN-RS, STRAIN-RT, STRAIN-SS, FE_EFFECTIVE_STRESS

Plane stress iso-beam, elastic-isotropic material: STRESS-RR, STRESS-RS, STRESS-SS, STRAIN-RR, STRAIN-RS, STRAIN-SS, FE_EFFECTIVE_STRESS

Plane strain iso-beam or axisymmetric shell, elastic-isotropic material: STRESS-RR, STRESS-RS, STRESS-TT, STRESS-SS, STRAIN-RR, STRAIN-RS, STRAIN-TT, STRAIN-SS, FE_EFFECTIVE_STRESS

General 3-D iso-beam, elastic-isotropic material, thermal effects: STRESS-RR, STRESS-RS, STRESS-RT, STRESS-SS, STRAIN-RR, STRAIN-RS, STRAIN-RT, STRAIN-SS, FE_EFFECTIVE_STRESS, THERMAL_STRAIN, ELEMENT_TEMPERATURE

Plane stress iso-beam, elastic-isotropic material, thermal effects:
STRESS-RR, STRESS-RS, STRESS-SS, STRAIN-RR,
STRAIN-RS, STRAIN-SS, FE_EFFECTIVE_STRESS,
THERMAL_STRAIN, ELEMENT_TEMPERATURE

Plane strain iso-beam or axisymmetric shell, elastic-isotropic
material, thermal effects: STRESS-RR, STRESS-RS,
STRESS-TT, STRESS-SS, STRAIN-RR, STRAIN-RS,
STRAIN-TT, STRAIN-SS, FE_EFFECTIVE_STRESS,
THERMAL_STRAIN, ELEMENT_TEMPERATURE

Thermo-isotropic material: STRESS (RST) ,
FE_EFFECTIVE_STRESS, STRAIN (RST) ,
THERMAL_STRAIN (RST) , ELEMENT_TEMPERATURE

Plastic-bilinear, plastic-multilinear materials: PLASTIC_FLAG,
STRESS (RST) , STRAIN (RST) ,
PLASTIC_STRAIN (RST) , THERMAL_STRAIN (RST) ,
FE_EFFECTIVE_STRESS, YIELD_STRESS,
ACCUM_EFF_PLASTIC_STRAIN, ELEMENT_TEMPERATURE

SMA: EFFECTIVE_STRESS, STRESS-RR, STRESS-RS,
STRESS-RT, STRESS-SS, STRESS-TT,
ACCUM_EFF_TRANSF_STRAIN, AUSTENITE_FRACTION,
DETWINNED_MARTENSITE_FRACTION, STRAIN-RR,
STRAIN-RS, STRAIN-RT, STRAIN-SS, STRAIN-TT,
THERMAL_STRAIN-RR, THERMAL_STRAIN-SS,
THERMAL_STRAIN-TT, TRANSFORMATION_STRAIN-RR,
TRANSFORMATION_STRAIN-RS,
TRANSFORMATION_STRAIN-RT,
TRANSFORMATION_STRAIN-SS,
TRANSFORMATION_STRAIN-TT,
TWINNED_MARTENSITE_FRACTION, SMA_FLAG

Thermo-plastic, creep, plastic-creep, multilinear-plastic-creep,
creep-variable, plastic-creep-variable, multilinear-plastic-creep-
variable materials: PLASTIC_FLAG,
NUMBER_OF_SUBINCREMENTS, STRESS (RST) ,
FE_EFFECTIVE_STRESS,
STRAIN (RST) , YIELD_STRESS,
PLASTIC_STRAIN (RST) ,
ACCUM_EFF_PLASTIC_STRAIN, CREEP_STRAIN (RST) ,

THERMAL_STRAIN(RST), ELEMENT_TEMPERATURE,
EFFECTIVE_CREEP_STRAIN

In the above lists,

STRESS(RST) = STRESS-RR, STRESS-RS,
STRESS-RT, STRESS-TT, STRESS-SS

with similar definitions for the other abbreviations used above.

See Section 13.1.1 for the definitions of those variables that are not self-explanatory.

ref. KJB
Section 6.3

Nodal forces: Nodal point forces are obtained using the relation

$${}^{t+\Delta t}\mathbf{F} = \int_{{}^{t+\Delta t}V} {}^{t+\Delta t}\mathbf{B}^T {}^{t+\Delta t}\boldsymbol{\tau} dV$$

where ${}^{t+\Delta t}\mathbf{B}$ is the strain-displacement matrix, the stresses are stored in ${}^{t+\Delta t}\boldsymbol{\tau}$, ${}^{t+\Delta t}V$ represents the volume, and the superscript $t + \Delta t$ refers to the conditions at time (load step) $t + \Delta t$.

These forces are accessible in the AUI using the variable names NODAL_FORCE-R, NODAL_FORCE-S, NODAL_FORCE-T, NODAL_MOMENT-R, NODAL_MOMENT-S, NODAL_MOMENT-T.

The end forces/moments are computed at the element local nodes. In the AUI, element local nodes are defined as element points of type label. For example, to access the result computed at element 5, local node 2, define an element point of type label with element number 5, label number 2.

2.6 Plate elements

- It is recommended to use shell elements (see Section 2.7) for analysis of flat shells, i.e., for analysis of plates.
- The plate element is kept in ADINA for backward compatibility only. For more information on the plate element, please see the ADINA Theory and Modeling Guides preceding ADINA 9.0.

2.7 Shell elements

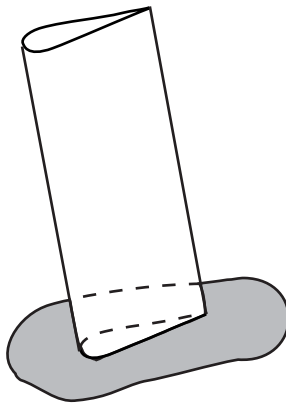
- The shell element is a 4- to 32-node (degenerate) isoparametric element that can be employed to model thick and thin general shell structures (see Fig. 2.7-1). However, depending on the application, the appropriate number of nodes on the element must be employed (see Fig. 2.7-2 and Section 2.7.10).
- The shell element used in ADINA is also applicable for analysis of flat shells, i.e., for analysis of plates.

2.7.1 Basic assumptions in element formulation

*ref. KJB
Sections 5.4.2
and 6.5.2*

- The basic equations used in the formulation of the element are given in ref. KJB.
- The shell element is formulated treating the shell as a three-dimensional continuum with the following two assumptions used in the Timoshenko beam theory and the Reissner/Mindlin plate theory:

Analysis of a turbine blade:



Analysis of a shell roof:

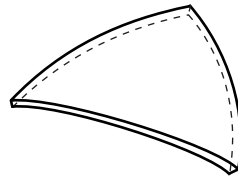


Figure 2.7-1: Some possible applications of shell elements

Assumption 1: Material particles that originally lie on a straight line "normal" to the midsurface of the structure remain on that straight line during the deformations.

Assumption 2: The stress in the direction normal to the midsurface of the structure is zero.

For the Timoshenko beam theory, the structure is the beam, and for the Reissner/Mindlin plate theory, the structure is the plate under consideration. In shell analysis, these assumptions correspond to a very general shell theory.

- In the calculations of the shell element matrices the following geometric quantities are used:
 - ▶ The coordinates of the node k that lies on the shell element midsurface at ${}^t x_k, {}^t y_k, {}^t z_k$ (see Fig. 2.7-3); (the left superscript denotes the configuration at time t)
 - ▶ The director vectors ${}^t \mathbf{V}_n^k$ pointing in the direction "normal" to the shell midsurface at node k

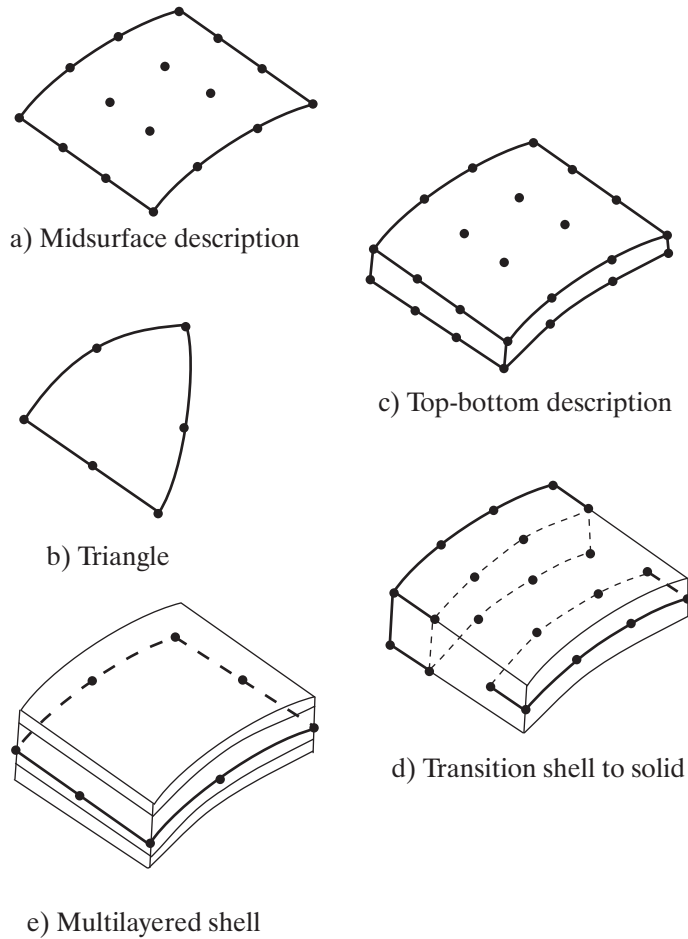


Figure 2.7-2: Examples of shell elements

ref. KJB
Fig. 5.33
page 437

- The shell thickness, a_k , at the nodal points measured in the direction of the director vectors ${}^t\mathbf{V}_n^k$ (see Fig. 2.7-4 and Section 2.7.2).

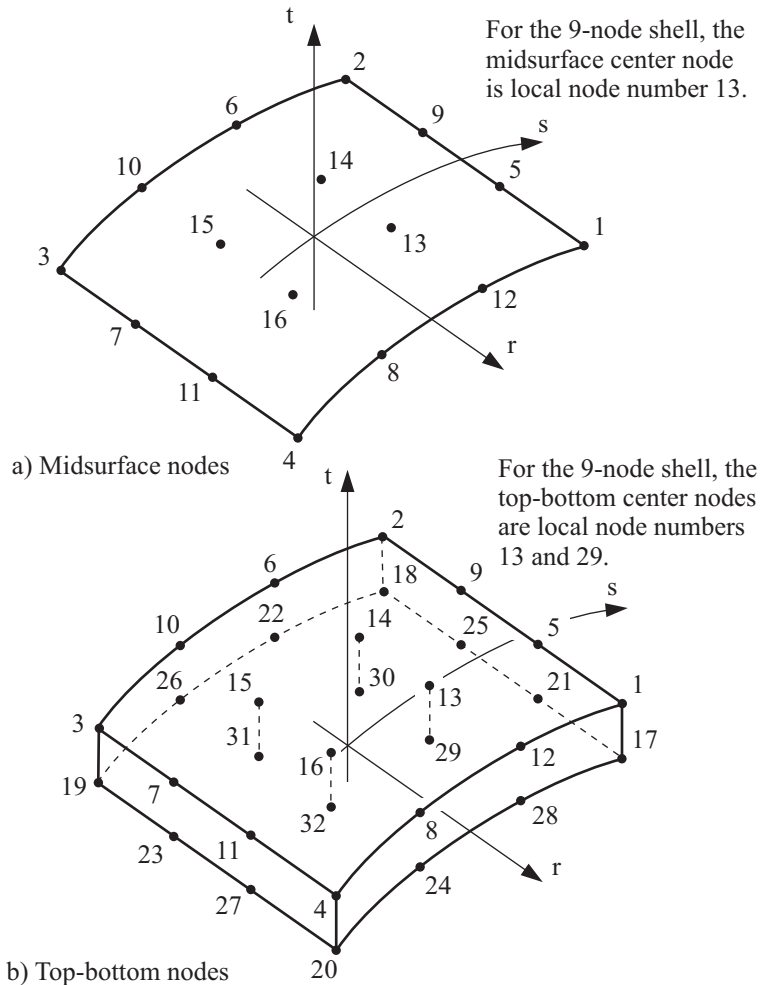


Figure 2.7-3: Some conventions for the shell element; local node numbering; natural coordinate system

- Fig. 2.7-4 shows a 4-node shell element with the shell midsurface nodal points and the nodal point director vectors. Using the midsurface nodal point coordinates, the shell midsurface is interpolated using the interpolation functions $h_k(r, s)$ given in ref. KJB. Similarly, using the vectors ${}^t\mathbf{V}_n$, the director vector ${}^t\mathbf{V}_n$ at any point P on the midsurface is obtained by interpolation using

the functions $h_k(r, s)$. Hence, with the shell thickness at nodal point k equal to a_k and the isoparametric coordinate t measured in the direction of ${}^t\mathbf{V}_n$, the geometry of the shell at any time t is defined by

$$\begin{aligned} {}^tx &= \sum_{k=1}^q h_k {}^tx_k + \frac{t}{2} \sum_{k=1}^q a_k h_k {}^tV_{nx}^k \\ {}^ty &= \sum_{k=1}^q h_k {}^ty_k + \frac{t}{2} \sum_{k=1}^q a_k h_k {}^tV_{ny}^k \\ {}^tz &= \sum_{k=1}^q h_k {}^tz_k + \frac{t}{2} \sum_{k=1}^q a_k h_k {}^tV_{nz}^k \end{aligned}$$

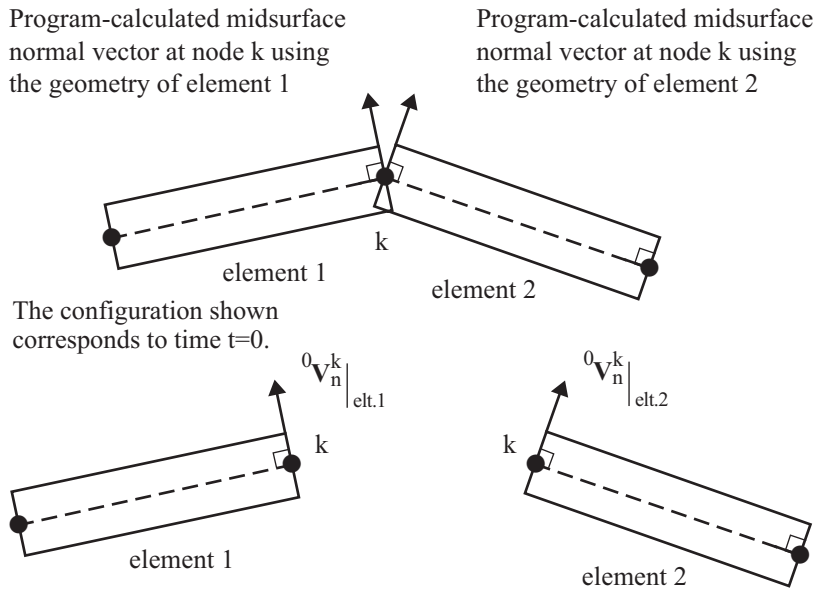
shell midsurface interpolation for material
interpolation points not on the midsurface

where q is the number of element nodes and ${}^tV_{nx}^k, {}^tV_{ny}^k, {}^tV_{nz}^k$ are the direction cosines of the shell director vector ${}^t\mathbf{V}_n^k$.

- The direction vectors at nodes can be either directly input or automatically generated by the program. When they are generated by the program, they can be created as the normal vectors of a geometrical surface, or as the averaged vectors of the surrounding elements, which may not be exactly normal to the corresponding geometric surface.
- The assumption 1 on the kinematic behavior of the shell enters the finite element solution in that the particles along the director vector ${}^t\mathbf{V}_n$ (interpolated from the nodal point director vectors ${}^t\mathbf{V}_n^k$) remain on a straight line during deformation.

Note that in the finite element solution, the vector ${}^t\mathbf{V}_n$ is not necessarily exactly normal to the shell midsurface. Figure 2.7-5(a) demonstrates this observation for a very simple case, considering the shell initial configuration. Furthermore, even if ${}^t\mathbf{V}_n$ is originally normal to the shell midsurface, after deformations have

taken place this vector will in general not be exactly perpendicular to the midsurface because of shear deformations (see Fig. 2.7-5(b)).



(a) Program-calculated normal vectors (two such vectors) at node k . These vectors are used as director vectors for the respective elements. Node k has 6 DOFs.

Figure 2.7-4: Convention for shell element thickness

ref. KJB
Section 5.4.2
page 440

- The assumption 2 on the stress situation enters the finite element solution in a manner that is dependent on the formulation employed:

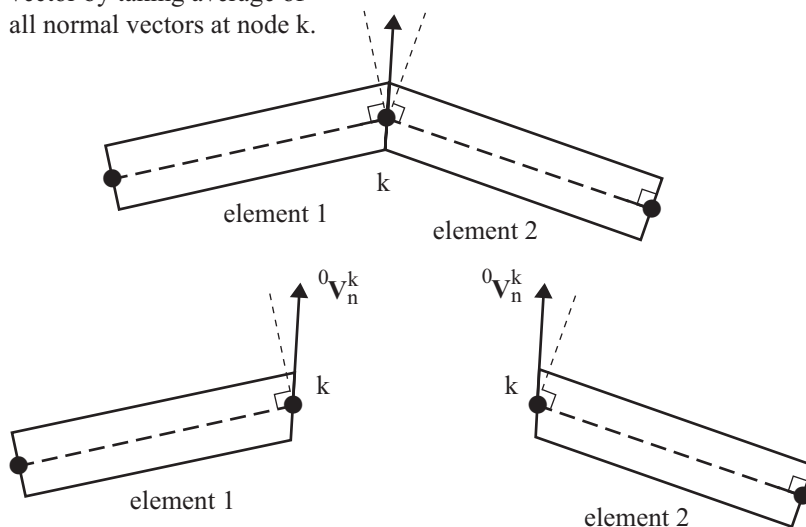
All formulations except for the large displacement/large strain formulations: The stress in the t -direction (i.e., in the direction of ${}^t\mathbf{V}_n$) is imposed to be zero. This is achieved by using the stress-strain relationship in the \bar{r}, \bar{s}, t coordinate system, shown in Fig. 2.7-6(a), with the condition that the stress in the direction t is zero.

Large displacement/large strain formulations: The stress in the \hat{t} -direction (not necessarily in the direction of ${}^t\mathbf{V}_n$) is imposed to be zero. This is achieved by using the stress-strain relationship in

the $\hat{r}, \hat{s}, \hat{t}$ coordinate system, shown in Fig. 2.7-6(b), with the condition that the stress in the direction \hat{t} is zero.

- Note that if the nodal point midsurface director vectors are input or generated (see Section 2.7.3) to be initially "exactly" normal to the midsurface of the shell element (the coordinates of this midsurface are obtained by interpolation from the nodal points on the midsurface), then assumption 1 is also exactly satisfied in the finite element solution, but assumption 2 is only approximately fulfilled in geometric nonlinear analysis, because the directions of the initially normal vectors are updated.

Program calculates director vector by taking average of all normal vectors at node k.



b) Program-calculated director vector (single vector) at node k. Node k has 5 DOFs.

Figure 2.7-4: (continued)

ref. KJB
pp. 399, 440

- The transverse shear deformations are assumed by default to be constant across the shell thickness. The use of the correction factor of 5/6 can be specified to improve the prediction of the displacement response, for the linear elastic and linear orthotropic models.

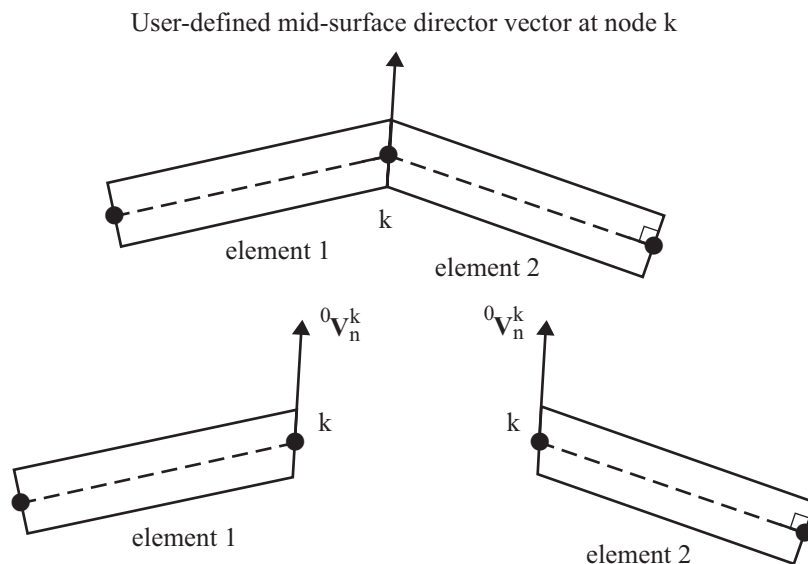
- The interpolation of the geometry of the shell element is always as described above, but for a specific solution time the current coordinates of the midsurface nodal points are used, and the current director vectors are employed. The midsurface nodal point coordinates are updated by the translational displacements of the nodes and the director vectors are updated using the rotations at the nodes (rotation increments in large displacement analysis).

- Incompatible modes can be used in conjunction with the 4-node shell element. The theory used is analogous to the theory used for the 2-D solid element, see Section 2.2.1 and ref KJB, Section 4.4.1. The incompatible modes are added to the midsurface displacement interpolations; only the membrane action of the 4-node shell element is affected.

A typical use of the incompatible modes feature would be to improve the in-plane bending response of the 4-node element.

The incompatible modes feature is only available for 4-node single layer shell elements in which all nodes are on the midsurface of the element. The incompatible modes feature is available in linear and nonlinear analysis, for all formulations.

- The thermal strain calculations in shell elements depend upon the shell element type. For the MITC3, MITC4, MITC6, MITC9, and MITC16 single layer shell elements, the thermal strain calculations give zero stresses for a uniform unrestrained thermal expansion. For the other shell elements, the thermal strain calculations can give non-zero stresses for a uniform unrestrained thermal expansion of a curved shell structure. As the mesh is refined, the stress calculations are improved.



c) User input director vector (single vector)
at node k. Node k has 5 DOFs.

Figure 2.7-4: (continued)

2.7.2 Material models and formulations

- The shell element can be used with the following material models: **elastic-isotropic, elastic-orthotropic, nonlinear-elastic, plastic-bilinear, plastic-multilinear, plastic-orthotropic, thermo-isotropic, thermo-orthotropic, thermo-plastic, creep, plastic-creep, multilinear-plastic-creep, creep-variable, plastic-creep-variable, multilinear-plastic-creep-variable, viscoelastic, plastic-cyclic.**
- The shell element can be used with a small displacement/small strain, large displacement/small strain, large displacement/large strain ULJ formulation or a large displacement/large strain ULH formulation. The small displacement/small strain and the large displacement/small strain formulations can be used with any of the above material models. The large displacement/large strain ULJ formulation can be used with the plastic-bilinear, plastic-multilinear, plastic-orthotropic or plastic-cyclic material models. The large displacement/large strain ULH formulation can be used

with the plastic-bilinear, plastic-multilinear or plastic-cyclic material models.

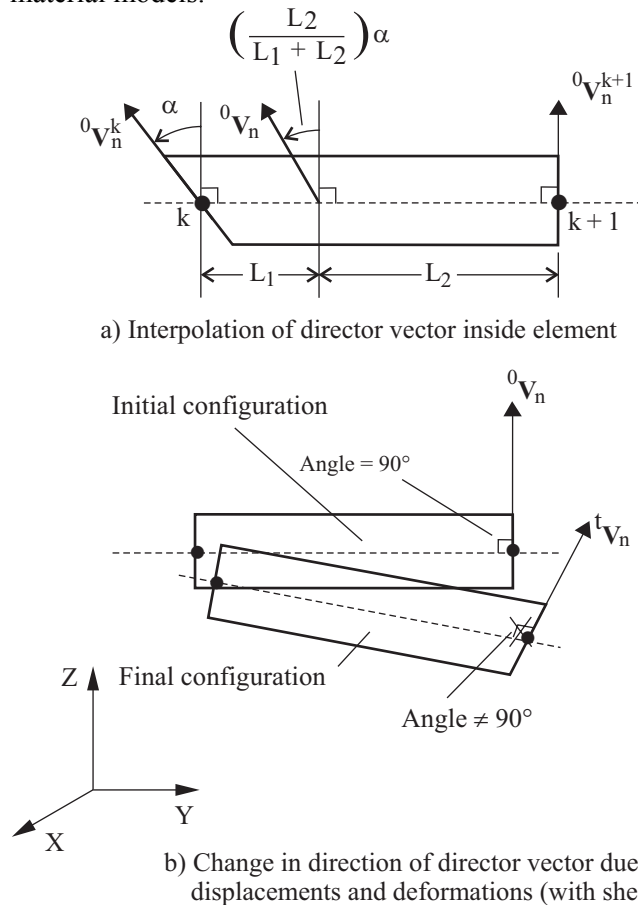


Figure 2.7-5: Example of change in direction of director vector due to deformations. At time t the vector ${}^t\mathbf{V}_n$ is not normal to the shell midsurface.

In the small displacement/small strain formulation, the displacements and rotations are assumed to be infinitesimally small. In the large displacement/small strain formulation, the displacements and rotations can be large, but the strains are assumed to be small. In the large displacement/large strain ULJ formulation, the total strains can be large, but the incremental strain for each time step should be small ($< 1\%$). In the large displacement/large strain ULH formulation, the total strains and

also the incremental strains can be large.

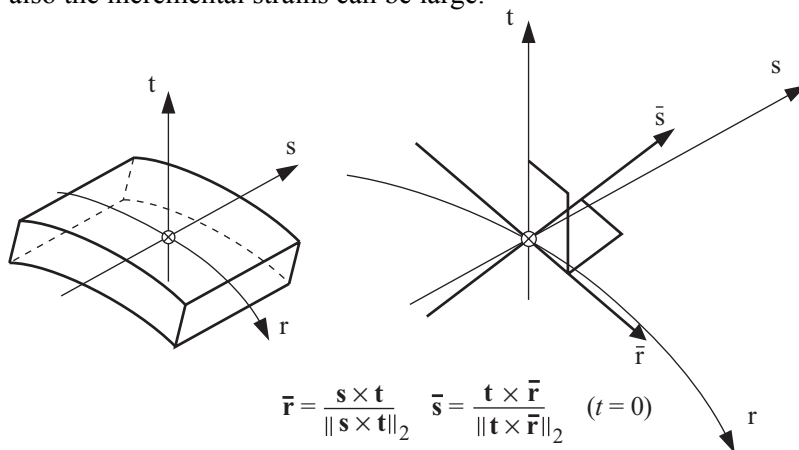


Figure 2.7-6a: Definition of the local Cartesian system $(\bar{\mathbf{r}}, \bar{\mathbf{s}}, \mathbf{t})$ at an integration point

The large displacement/large strain formulation can only be used in conjunction with 3-node, 4-node, 6-node, 9-node or 16-node single layer shell elements, in which the shell geometry is described in terms of midsurface nodes.

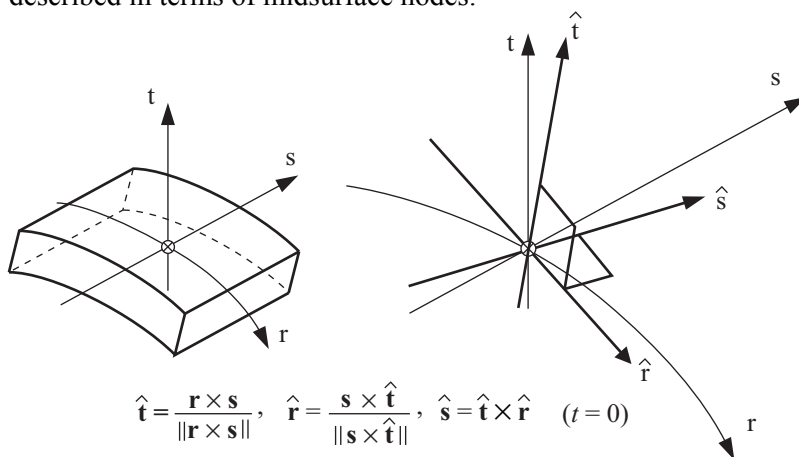


Figure 2.7-6b: Definition of the midsurface Cartesian system $(\hat{\mathbf{r}}, \hat{\mathbf{s}}, \hat{\mathbf{t}})$ at an integration point

The use of a linear material with the small displacement/small strain formulation corresponds to a linear formulation, and the use of a nonlinear material with the small displacement/small strain formulation corresponds to a materially-nonlinear-only formulation.

- The effects of displacement-dependent pressure loading are taken into account in the large displacement formulation.

In frequency analysis, the stiffness matrix is optionally updated by deformation-dependent pressure loads acting onto shell elements. The update can improve the obtained frequencies and mode shapes when the shell structure is relatively flexible and deformation-dependent pressure loads are applied.

2.7.3 Shell nodal point degrees of freedom

- Either 5 or 6 degrees of freedom can be assigned at a shell element midsurface node.

5 degrees of freedom node: A node " k " that is assigned 5 degrees of freedom incorporates the following assumptions:

- ▶ The translations u_k, v_k, w_k are referred to the global Cartesian system (or to the skew system if such a system is defined at the node)

- ▶ Only one director vector (denoted at time = 0 as ${}^0\mathbf{V}_n^k$) is associated with the node. You can directly enter the director vector or let the program calculate it (see Fig. 2.7-4). In the latter case, the program calculates the director vector by taking the average of all normal vectors (one normal vector is generated per shell element attached to node k) at the node.

If two (or more) elements attached to the node have oppositely directed normals, ADINA reverses the oppositely directed normals, so that all normals attached to the node have (nearly) the same direction.

- ▶ The rotations α_k, β_k are referred to the local midsurface system (see Fig. 2.7-7) defined at time = 0 by

$${}^0\mathbf{V}_1^k = \frac{\mathbf{Y} \times {}^0\mathbf{V}_n^k}{\|\mathbf{Y} \times {}^0\mathbf{V}_n^k\|_2}$$

$${}^0\mathbf{V}_2^k = {}^0\mathbf{V}_n^k \times {}^0\mathbf{V}_1^k$$

For the special case when the ${}^0\mathbf{V}_n^k$ vector is parallel to the Y axis, the program uses the following conventions:

$${}^0\mathbf{V}_1^k \equiv \mathbf{Z} \quad {}^0\mathbf{V}_2^k \equiv \mathbf{X} \quad \text{when} \quad {}^0\mathbf{V}_n^k \equiv +\mathbf{Y}$$

and

$${}^0\mathbf{V}_1^k \equiv -\mathbf{Z} \quad {}^0\mathbf{V}_2^k \equiv \mathbf{X} \quad \text{when} \quad {}^0\mathbf{V}_n^k \equiv -\mathbf{Y}$$

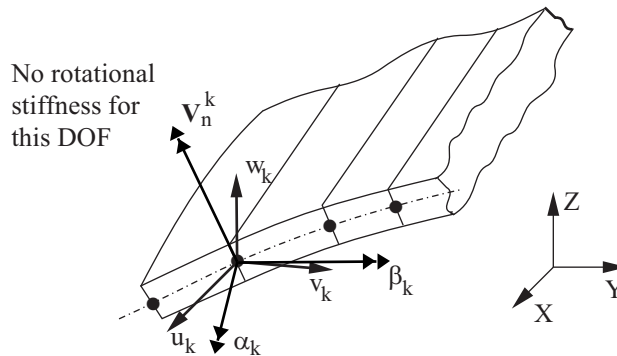


Figure 2.7-7: Shell degrees of freedom at node k

When using the large displacement formulation, the definitions of ${}^0\mathbf{V}_1^k$ and ${}^0\mathbf{V}_2^k$ are only used at time=0 (in the initial configuration) after which the vectors ${}^t\mathbf{V}_n^k$ and ${}^t\mathbf{V}_1^k$ are updated using incremental rotations at the nodal points, and ${}^t\mathbf{V}_2^k$ is calculated by the cross-product ${}^t\mathbf{V}_2^k = {}^t\mathbf{V}_n^k \times {}^t\mathbf{V}_1^k$.

- The rotational degree of freedom along ${}^0\mathbf{V}_n^k$ is automatically deleted by ADINA.

6 degrees of freedom node: A node " k " that is assigned 6 degrees of freedom incorporates the following assumptions:

- ▶ The translations u_k , v_k , w_k are referred to the global Cartesian system or any skew system assigned to node k .
 - ▶ A director vector would in general not be input for this node. (Note that any director vector input is ignored for a shell node with six degrees of freedom.)
 - ▶ The program generates as many normal vectors at node k as there are shell elements attached to the node (see Fig. 2.7-4(a)). Hence each individual shell element establishes at node k a vector normal to its midsurface. The components of the shell element matrices corresponding to the rotational degrees of freedom at this node are first formulated in the local midsurface system defined by the normal vector and then rotated to the global Cartesian system (or to the skew system, if defined at the node).
 - ▶ The three rotational degrees of freedom at node k referred to the global Cartesian system (or to the skew system, if defined at the node) can be free or be deleted.
- Some modeling recommendations are given below for the use of the shell elements:
 - ▶ Always let all director vectors be established by the program; director vectors should only be defined via input if specific director vectors are required in the modeling.
 - ▶ Always specify 5 degrees of freedom at all shell midsurface nodes except for the following cases in which 6 degrees of freedom should be used:
 - (i) shell elements intersecting at an angle,
 - (ii) coupling of shell elements with other types of structural elements such as isoparametric beams (e.g., in the modeling of stiffened shells),
 - (iii) coupling of rigid links (see Section 5.15.2) to the shell midsurface nodes,

- (iv) coupling of constraints or generalized constraints to the shell midsurface nodes,
- (v) imposing rotational moments or boundary conditions at the node.
- (vi) coupling of alignment elements that have alignment properties to the shell midsurface nodes

All of the above considerations are illustrated in the schematic example shown in Fig. 2.7-8. Note that in the example, the use of 6 degrees of freedom at node 1 together with the deletion of the X-rotation degree of freedom is only applicable in small displacement analysis.

If a large displacement formulation is used in this example, node 1 must be assigned 5 degrees of freedom and the applied concentrated moments must then refer to the local midsurface system at that node, which consists of the global directions for u_k , v_k , w_k and the local directions ${}^t\mathbf{V}_1^k$ and ${}^t\mathbf{V}_2^k$ for the incremental rotations α_k and β_k . (Alternatively, 6 degrees of freedom could be used if a beam of very small stiffness is defined along the edge of the plate.)

There are a number of implications regarding the use of 6 DOF nodes on flat or nearly flat shells. These implications arise because there is no rotational stiffness in the ${}^t\mathbf{V}_n$ direction.

1) No attached structural element/rigid link (Fig. 2.7-9(a))

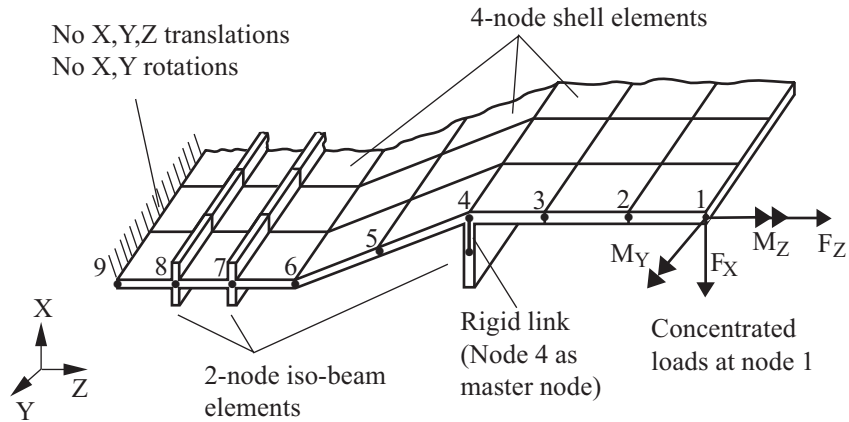
a) There will be a zero pivot in the stiffness matrix. If a moment is applied into the ${}^t\mathbf{V}_n$ direction, the rotation in the ${}^t\mathbf{V}_n$ direction will be infinite. In addition, the moment applied into the ${}^t\mathbf{V}_n$ direction will not cause equilibrating reactions at the fixities.

*2) Structural element/rigid link attached to the node
(Fig 2.7-9(b)(-d)).*

a) If the structural element/rigid link is unsupported, there will be a zero pivot in the global stiffness matrix. Even if the structural element/rigid link is supported, there may be a zero pivot in the global stiffness matrix, depending upon how the structural

element/rigid link is supported.

b) Moments in the tV_n direction will not be transmitted from the structural element or rigid link into/from the shell element.



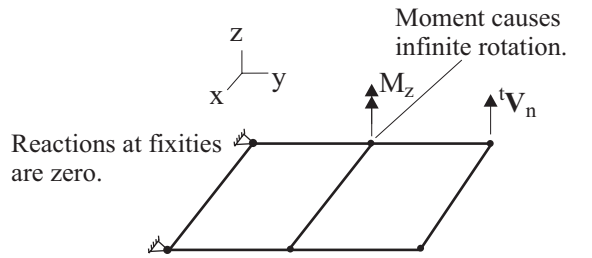
Node	Number of DOF	Degree of freedom						Reference system
		1	2	3	4	5	6	
1	6	✓	✓	✓	—	✓	✓	Global
2	5	✓	✓	✓	✓	✓	—*	Midsurface
3	5	✓	✓	✓	✓	✓	—*	Midsurface
4	6	✓	✓	✓	✓	✓	✓	Global
5	5	✓	✓	✓	✓	✓	—*	Midsurface
6	6	✓	✓	✓	✓	✓	✓	Global
7	6	✓	✓	✓	✓	✓	✓	Global
8	6	✓	✓	✓	✓	✓	✓	Global
9	6	—	—	—	—	—	✓	Global

— = Deleted because of no stiffness at this DOF

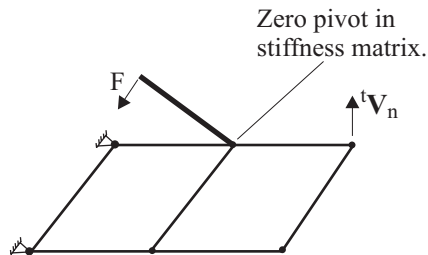
✓ = Free DOF

* = The AUI automatically deletes the 6th DOF when a midsurface system is specified for the node

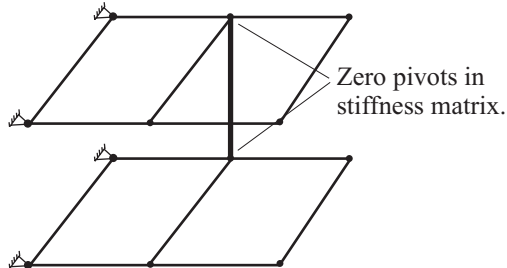
Figure 2.7-8: Example on the recommended use of shell elements



a) Moment applied in shell normal direction.



b) Structural element/rigid link attached to node, structural element/rigid link is unsupported.



c) Structural element/rigid link attached to node, structural element/rigid link is perpendicular to shell elements

Figure 2.7-9: Flat shell with 6 DOFs at a node

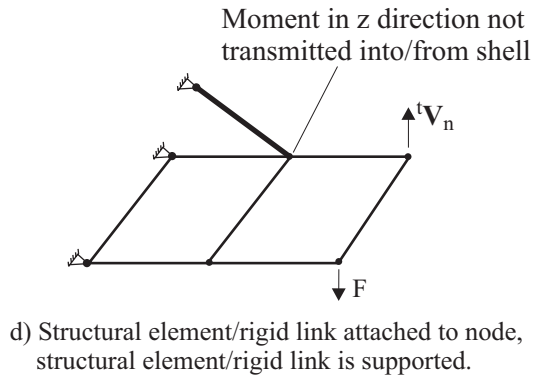
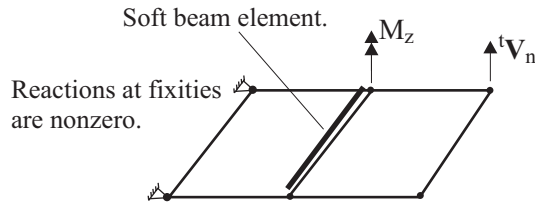


Figure 2.7-9: (continued)

ADINA has the option of applying a “drilling stiffness” to 6 DOF nodes on flat shells.

The AUI command is MASTER SINGULARITY-STIFFNESS=YES (on by default). When the drilling stiffness option is used, any moment applied to the node into the tV_n direction is taken by the drilling stiffness rotational spring (which is grounded). The drilling stiffness spring eliminates the zero pivot in the stiffness matrix, hence in some cases, a solution can be obtained. But any moment applied into the tV_n direction will still not cause equilibrating reactions at the fixities. And using a drilling stiffness value that is too large may lead to an incorrect solution.

An alternative to using the drilling stiffness option is to connect the 6 DOF nodes on flat shells to neighboring shell nodes using soft beam elements (so-called “weld elements”). This idea is shown in Fig 2.7-9(e). Then moments applied into the tV_n direction will be taken by the weld elements, and these moments will cause equilibrating reactions at the fixities. The weld elements also provide stiffness in the tV_n direction, so that there will be no zero pivots.



e) Soft beam element takes applied moment.

Figure 2.7-9: (continued)

2.7.4 Transition elements

- The shell elements can also be employed as transition elements. A transition element is obtained by using, instead of a shell midsurface node, one node on the top surface and one node on the bottom surface of the shell element. Each of these nodes has then only 3 degrees of freedom corresponding to translations in the global X, Y, Z coordinate directions. Fig. 2.7-10(a) shows a cubic transition element.
- Note that although the degrees of freedom at a shell transition node are those of a three-dimensional isoparametric element, the assumption that the stress "normal" to the midsurface of the shell (in the t-direction) is zero distinguishes a transition element from a 3-D solid element even when a shell element has only transition nodes.

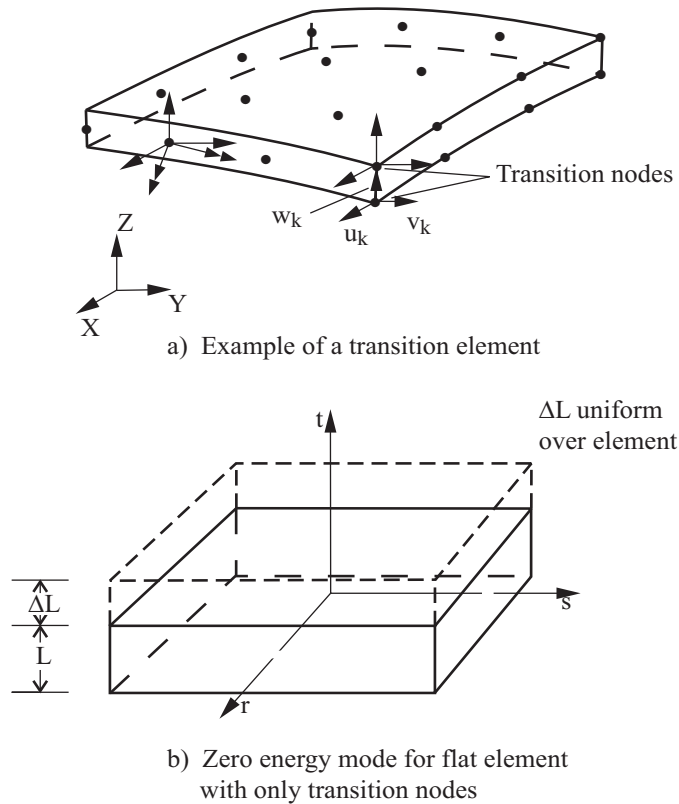


Figure 2.7-10: Transition elements

- In particular, if only transition nodes are used on the element, then the element has a zero energy mode corresponding to a uniform strain in the t -direction. Fig. 2.7-10(b) shows this zero-energy mode. For a 3-D solid element, this mode corresponds, of course, to a uniform strain in the t -direction.

If not all nodes on a transition element are transition nodes, i.e., there is at least one midsurface node, then the zero-energy mode shown in Fig. 2.7-10(b) is not present.

- The transition element is particularly useful in modeling shell and shell to solid intersections, see Fig. 2.7-11 and the following reference:

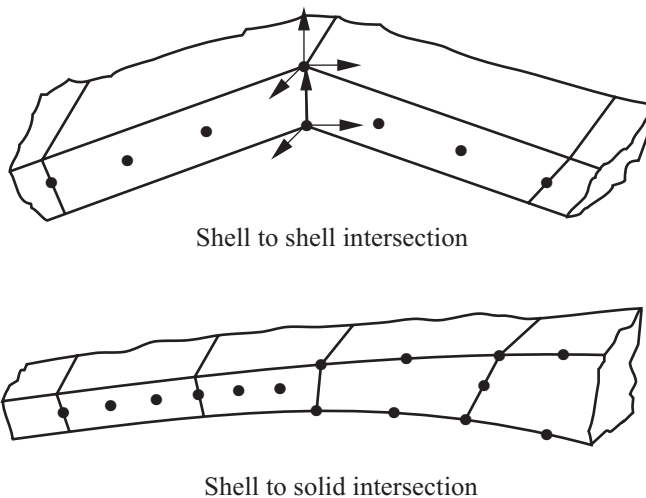


Figure 2.7-11: Examples of use of transition element

ref. K.J. Bathe and L.W. Ho, "Some Results in the Analysis of Thin Shell Structures," in *Nonlinear Finite Element Analysis in Structural Mechanics*, W. Wunderlich et al (eds.), Springer-Verlag, 1981.

2.7.5 Numerical integration

ref. KJB
Section 5.5

- Numerical integration is used for the evaluation of the element matrices, and the default integration or a higher order should always be used; then no spurious zero energy modes are present.
- Gauss numerical integration is used in the r-s plane. For the 4-node element, the default order is 2×2 integration. For the 8- and 9-node elements, the default order is 3×3 integration, and for the 16-node elements, the default order is 4×4 integration.

ref. KJB
Section 6.8.4

- Either Gauss, Newton-Cotes or trapezoidal rule numerical integration is used through the shell thickness. For Gauss integration, 2 to 6 points through the thickness can be selected. For Newton-Cotes integration, 3, 5 or 7 points through the thickness can be selected. For trapezoidal rule integration, 2 to 20 points through the thickness can be selected. Fig 2.7-12 shows some examples.

Usually, 2-point Gauss or 3-point Newton-Cotes integration is appropriate for an elastic material, but a higher integration order may be more effective for elastic-plastic analysis.

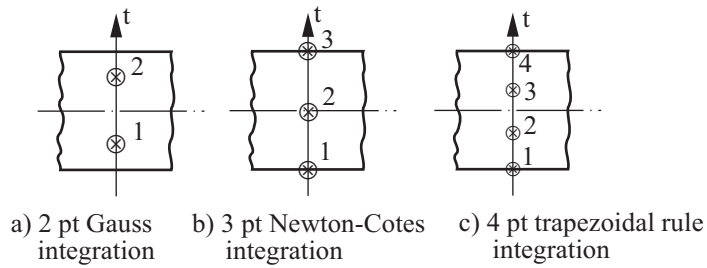
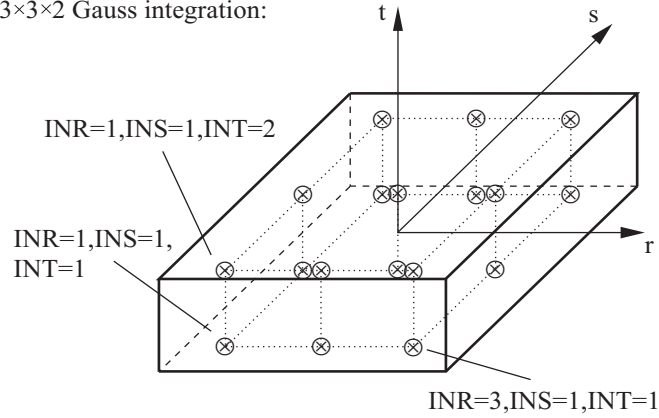


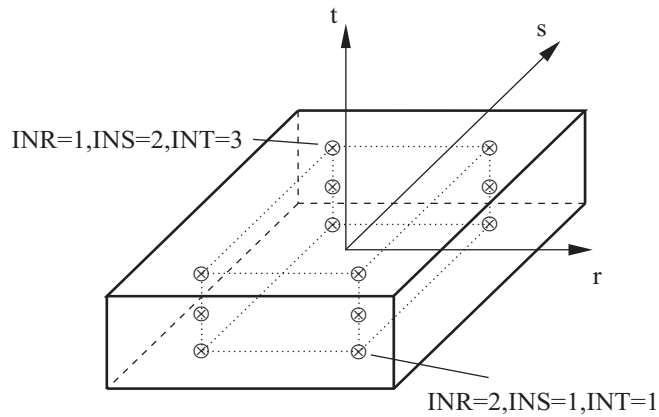
Figure 2.7-12: Examples of numerical integration through the shell element thickness

- The labeling of the integration points for quadrilateral and triangular elements is given in Fig. 2.7-13. The locations of the integration points in the r - s plane are the same as for the 2-D solid elements (see Section 2.2.3).

3×3×2 Gauss integration:



2×2×3 Newton-Cotes integration:



a) Examples of integration point labeling for quadrilateral shell elements

Figure 2.7-13: Examples of integration point labeling

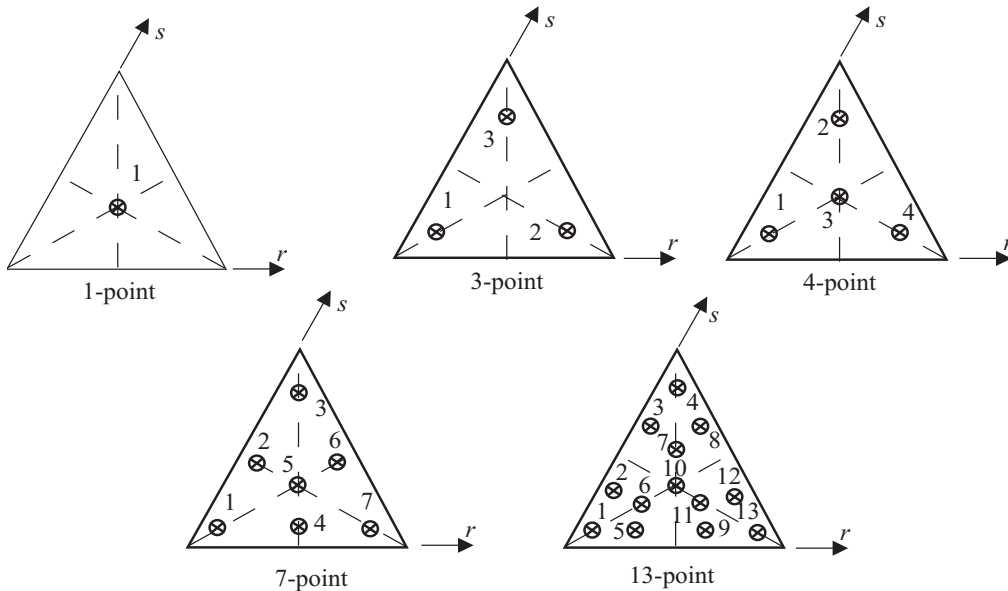
b. Integration points position on r - s plane for the triangular shell elements

Figure 2.7-13: (continued)

- The default number of Gauss integration points in the r - s plane for the MITC3 element is 3; for the MITC6 element is 7; for the collapsed MITC4 triangular element is 4; for the collapsed DISP7 element is 7; and for the collapsed DISP10 element is 13 (see also Fig. 2.7-18).

2.7.6 Composite shell elements

- The composite shell elements are kinematically formulated in the same way as the single layer shell elements, but
 - ▶ An arbitrary number of layers can be used to make up the total thickness of the shell.
 - ▶ Each layer can be assigned one of the different material models available. The element is nonlinear if any of the material models is nonlinear, or if the large displacement formulation is used.
 - ▶ The computation of accurate transverse shear stresses based on a three-dimensional theory can be requested. This way the

condition of zero transverse shear stresses at the top and bottom surfaces of the shell are satisfied, and the transverse shear stresses are continuous at layer interfaces.

- Layers are numbered in sequential order starting from 1 at the bottom of the shell.
- The layer thicknesses can be assigned using one of two general approaches:
 - Specify the total element thickness and the percentage of thickness for each layer.
 - Specify the data of each layer (or ply) in terms of the weight per unit surface of the fiber W_f , the density of the fiber ρ_f and the fiber volume fraction of the fiber-matrix compound ϕ_f .

The AUI then computes the thickness of each layer

automatically using the formula $h = \frac{W_f}{\rho_f \phi_f}$, in which it is

assumed that the ply is made of uniaxial fibers, so that the fiber thickness fraction perpendicular to the plate midsurface is proportional to the fiber volume fraction. The total thickness of the multilayered shell is then the sum of the ply thicknesses.

This approach is especially useful when the layers are fiber-matrix composites.

- In order to take into account the change of material properties from one layer to another, numerical integration of the mass and stiffness matrices is performed layer by layer using reduced natural coordinates through the thickness of the element (see Fig. 2.7-14 and 2.7-15). The relation between the element natural coordinate t and the reduced natural coordinate t^n of layer n is:

$$t = -1 + \frac{1}{a} \left[2 \left(\sum_{i=1}^n \ell^i \right) - \ell^n (1 - t^n) \right] \quad (2.7-1)$$

with

t = element natural coordinate through the thickness

t^n = layer n natural coordinate through the thickness

ℓ^i = thickness of layer i

a = total element thickness

a and ℓ^i are functions of r and s .

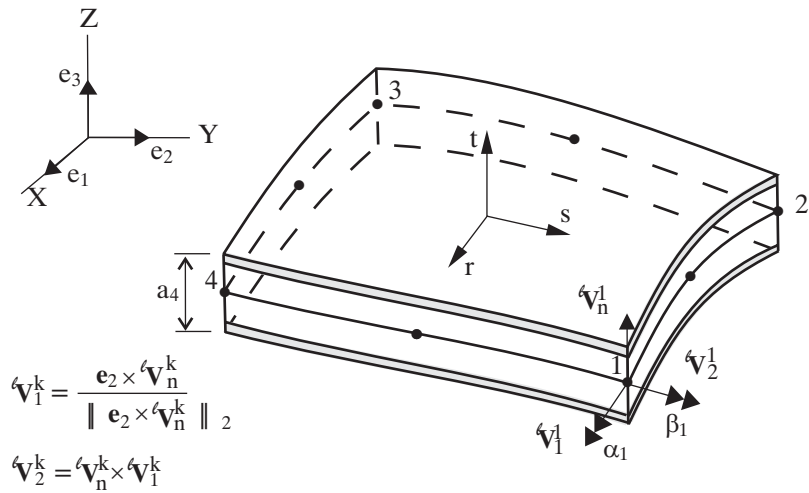


Figure 2.7-14: 8-node composite shell element

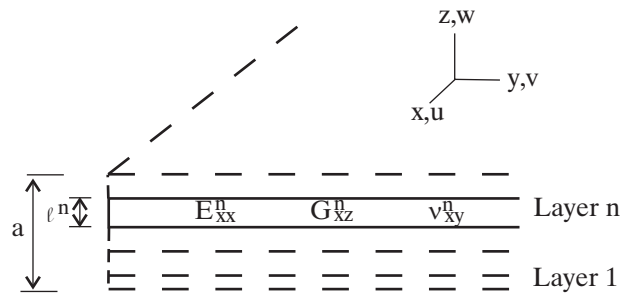


Figure 2.7-15: Multilayered plate

The geometry of layer n is given by:

$${}^\ell x_i = \sum_{k=1}^N h_k {}^\ell x_i^k + \sum_{k=1}^N \left[m_k^n + t^n \frac{\ell_k^n}{2} \right] h_k {}^\ell V_{ni}^k \quad (2.7-2)$$

with

- ${}^\ell x_i$ = coordinates of a point inside layer n
- N = number of nodes
- h_k = interpolation functions
- ${}^\ell x_i^k$ = Cartesian coordinates of node k
- ${}^\ell V_{ni}^k$ = components of normal vector ${}^\ell \mathbf{V}_n^k$
- a_k = total element thickness at node k
- ℓ_k^i = thickness of layer i at node k
- $\ell_{[]}$ = 0 in the initial configuration, 1 in the deformed configuration

In the above formula, the expression

$$m_k^n = -\frac{a_k}{2} + \sum_{i=1}^n \ell_k^i - \frac{\ell_k^n}{2} \quad (2.7-3)$$

corresponds to the distance at node k between the element midsurface and the midsurface of layer n .

Accordingly, the displacements in layer n are:

$$u_i = \sum_{k=1}^N h_k u_i^k + \sum_{k=1}^N \left(m_k^n + \frac{t^n \ell_k^n}{2} \right) \left(-{}^0 V_{2i}^k \alpha_k + {}^0 V_{1i}^k \beta_k \right) \quad (2.7-4)$$

with

- u_i^k = components of nodal displacements at node k
- α_k, β_k = rotations of ${}^0 \mathbf{V}_n^k$ about ${}^0 \mathbf{V}_1^k$ and ${}^0 \mathbf{V}_2^k$ (see Fig. 2.7-14)

Using the expressions of the coordinates and displacements

defined in Eqs. 2.7-2 and 2.7-4, the contribution of each layer to the element stiffness and mass matrices can be evaluated.

- In the analysis of thick plates and shells or in the analysis of multilayered structures such as composite sandwiches with low shear rigidity, the transverse shear deformation energy may not be negligible compared to the flexural energy and high transverse shear stresses may be encountered. In these cases, the shell kinematic formulation must be amended to obtain more accurate expressions of the shear deformation energy and transverse shear stresses.

By combining for a plate in cylindrical bending, the plate equilibrium equations and the 3-dimensional equilibrium equations, an expression of the shear stress which satisfies stress continuity at the layer interfaces and zero boundary values at the top and bottom surfaces of the shell can be obtained. Simultaneously, a shear reduction factor is obtained which can be used to improve the shear deformation energy approximation.

The expression of the shear factor is given by

$$k_{xz} = \frac{I_{xx}^2}{\int_{-a/2}^{a/2} G_{xz} dz \int_{-a/2}^{a/2} \frac{g_{xz}^2}{G_{xz}} dz} \quad (2.7-5)$$

and the transverse shear stress in terms of the shear strain value $\bar{\gamma}_{xz}$ is

$$\sigma_{xz}(z) = \bar{\gamma}_{xz} \frac{I_{xx} g_{xz}(z)}{\int_{-a/2}^{a/2} \frac{g_{xz}^2}{G_{xz}} dz} \quad (2.7-6)$$

where

$$I_{xx} = \int_{-a/2}^{a/2} D_{xx} (z - z_0)^2 dz$$

$$D_{xx}^n = \frac{E_{xx}^n}{1 - \nu_{xy}^n \nu_{yx}^n}$$

E_{xx} = Young's modulus

G_{xz} = shear modulus

ν_{xy}^n, ν_{yx}^n = Poisson's ratios

a = total thickness

$$g_{xz}(z) = - \int_{-a/2}^z D_{xx}^n(\xi - z_0) d\xi$$

Equations 2.7-5 and 2.7-6 are used in ADINA to calculate shear correction factors for each element and shear stress profiles at each integration point.

The shear correction factor calculation is only available for the linear isotropic elastic and linear orthotropic material models.

ref. O. Guillermin, M. Kojić, K.J. Bathe, "Linear and Nonlinear Analysis of Composite Shells," *Proceedings, STRUCOME 90, DATAID AS & I*, Paris, France, November 1990.

2.7.7 Mass matrices

- The shell element can be employed with a lumped or a consistent mass matrix, except for explicit dynamic analysis which always uses a lumped mass.
- The consistent mass matrix is calculated using the isoparametric formulation with the shell element interpolation functions.
- The lumped mass for translational degrees of freedom of midsurface nodes is M/n where M is the total element mass and n is the number of nodes. No special distributory concepts are employed to distinguish between corner and midside nodes, or to account for element distortion.

The rotational lumped mass for all except explicit dynamic analysis is, $\frac{M}{n} \cdot \frac{1}{12} (t_{av}^2)$ where t_{av} is the average shell thickness. The same rotational mass matrix is assumed for 5- and 6-degree of freedom nodes, and is applied to all rotational degrees of freedom.

The rotational lumped mass for explicit dynamic analysis is

$\frac{M}{n} \cdot \frac{1}{12} (t_{av}^2 + A)$, where t_{av} is the average shell thickness and A is the cross-sectional area. The rotational masses are scaled up to ensure that the rotational degrees of freedom do not reduce the critical time step for shell elements. The same rotational mass matrix is assumed for 5- and 6-degree of freedom nodes and is applied to all rotational degrees of freedom.

- The rotational lumped mass can be multiplied by a user-specified scalar ETA (except in explicit analysis).

2.7.8 Anisotropic failure criteria

- The study of material failure in composite structures has shown that various criteria can be used to assess failure. The following failure criteria are provided in ADINA for the analysis of shell structures using the elastic-isotropic, elastic-orthotropic, thermo-isotropic or thermo-orthotropic material models:
 - ▶ Maximum stress failure criterion
 - ▶ Maximum strain failure criterion
 - ▶ Tsai-Hill failure criterion
 - ▶ Tensor polynomial failure criterion
 - ▶ Hashin failure criterion
 - ▶ User-supplied failure criterion
- These criteria can be evaluated during the analysis to determine whether the material has failed. However, the stresses and material properties of the model are not changed as a consequence of this calculation. It is therefore intended only for use in linear elastic or thermo-elastic analysis, or as a stress state indicator in nonlinear analysis.
- The material failure is investigated at each integration point of each element during the analysis. The failure criteria values (and possibly modes) are printed and/or saved according to the ADINA stress printing and saving flags.
- The **maximum stress failure criterion** compares each component of the stress tensor (resolved in the material (abc))

directions) to maximum stress values. The input constants for this criterion are:

- X_t = maximum allowable tensile stress in the material a-direction
- X_c = maximum allowable compressive stress in the material a-direction
- Y_t = maximum allowable tensile stress in the material b-direction
- Y_c = maximum allowable compressive stress in the material b-direction
- Z_t = maximum allowable tensile stress in the material c-direction
- Z_c = maximum allowable compressive stress in the material c-direction
- S_{ab} = maximum allowable absolute shear stress in the a-b plane
- S_{ac} = maximum allowable absolute shear stress in the a-c plane
- S_{bc} = maximum allowable absolute shear stress in the b-c plane

The failure of the material occurs when any of the following inequalities is not satisfied anymore:

$$\begin{aligned} X_c &< \tau_a < X_t \\ Y_c &< \tau_b < Y_t \\ Z_c &< \tau_c < Z_t \\ -S_{ab} &< \tau_{ab} < S_{ab} \\ -S_{ac} &< \tau_{ac} < S_{ac} \\ -S_{bc} &< \tau_{bc} < S_{bc} \end{aligned}$$

where $(\tau_a, \tau_b, \tau_c, \tau_{ab}, \tau_{ac}, \tau_{bc})$ is the stress vector referred to the principal material directions.

The above conditions are applicable to 3-D analysis. In plane stress analysis, the parameters Z_b , Z_c , S_{ac} and S_{bc} are not used.

- Note that the maximum stress criterion indicates the mode of failure. Note that this criterion does not take into account any

interaction between failure modes.

- The **maximum strain failure criterion** compares each component of the strain tensor (resolved in the material (abc) directions) to maximum strain values. The input constants for this criterion are:

X_{et} = maximum allowable tensile strain in the material a-direction

X_{ec} = maximum allowable compressive strain in the material a-direction

Y_{et} = maximum allowable tensile strain in the material b-direction

Y_{ec} = maximum allowable compressive strain in the material b-direction

Z_{et} = maximum allowable tensile strain in the material third c-direction

Z_{ec} = maximum allowable compressive strain in the material c-direction

$S_{\varepsilon ab}$ = maximum allowable absolute shear strain in the a-b plane

$S_{\varepsilon ac}$ = maximum allowable absolute shear strain in the a-c plane

$S_{\varepsilon bc}$ = maximum allowable absolute shear strain in the b-c plane

The failure of the material occurs when any one of the following inequalities is not satisfied anymore:

$$\begin{aligned} X_{\varepsilon c} &< \varepsilon_a < X_{\varepsilon t} \\ Y_{\varepsilon c} &< \varepsilon_b < Y_{\varepsilon t} \\ Z_{\varepsilon c} &< \varepsilon_c < Z_{\varepsilon t} \\ -S_{\varepsilon ab} &< \gamma_{ab} < S_{\varepsilon ab} \\ -S_{\varepsilon ac} &< \gamma_{ac} < S_{\varepsilon ac} \\ -S_{\varepsilon bc} &< \gamma_{bc} < S_{\varepsilon bc} \end{aligned}$$

where $(\varepsilon_a, \varepsilon_b, \varepsilon_c, \gamma_{ab}, \gamma_{ac}, \gamma_{bc})$ is the strain vector referred to the principal material directions.

The above conditions are applicable to 3-D analysis. In plane

stress analysis, however, the parameters Z_{ct} , Z_{cs} , S_{cac} and S_{cbc} are not used.

- Note that the maximum strain criterion indicates the mode of failure. Note that this criterion does not take into account any interaction between failure modes.
- The **Tsai-Hill failure criterion** provides a yield criterion applicable to materials with anisotropic failure strengths. This is an extension of the von Mises yield criterion. Failure of the material occurs when the following inequality is not satisfied anymore:

$$(G + H)\tau_a^2 + (F + H)\tau_b^2 + (F + G)\tau_c^2 - 2H\tau_a\tau_b - 2G\tau_a\tau_c - 2F\tau_b\tau_c + 2L\tau_{bc}^2 + 2M\tau_{ac}^2 + 2N\tau_{ab}^2 < 1$$

with

$$F = \frac{1}{2} \left(-\frac{1}{X^2} + \frac{1}{Y^2} + \frac{1}{Z^2} \right)$$

$$G = \frac{1}{2} \left(\frac{1}{X^2} - \frac{1}{Y^2} + \frac{1}{Z^2} \right)$$

$$H = \frac{1}{2} \left(\frac{1}{X^2} + \frac{1}{Y^2} - \frac{1}{Z^2} \right)$$

$$L = \frac{1}{2 S_{bc}^2}$$

$$M = \frac{1}{2 S_{ac}^2}$$

$$N = \frac{1}{2 S_{ab}^2}$$

where

- X = maximum allowable absolute stress in the material a-direction
- Y = maximum allowable absolute stress in the material b-direction
- Z = maximum allowable absolute stress in the material c-direction

- S_{ab} = maximum allowable absolute shear stress in the a-b plane
 S_{ac} = maximum allowable absolute shear stress in the a-c plane
 S_{bc} = maximum allowable absolute shear stress in the b-c plane

In the case when plane stress conditions are used, the material behavior with respect to failure is considered as transversely isotropic ($Y=Z$), and the failure criterion reduces to

$$\frac{\tau_a^2}{X^2} - \frac{\tau_a \tau_b}{X^2} + \frac{\tau_b^2}{Y^2} + \frac{\tau_{ab}^2}{S_{ab}^2} < 1$$

In this case, the maximum allowable stresses Z , S_{bc} and S_{ac} are not used.

- Note that this criterion includes the effect of interactions between failure modes, but does not indicate a specific mode of failure.
- The **tensor polynomial failure criterion** assumes the existence of a failure surface in the stress space of the form:

$$F_i \tau_i + F_{ij} \tau_i \tau_j = 1 \quad \text{with } i, j = 1, \dots, 6$$

with the convention

$$\begin{aligned} \tau_1 &= \tau_a ; \tau_2 = \tau_b ; \tau_3 = \tau_c \\ \tau_4 &= \tau_{ab} ; \tau_5 = \tau_{ac} ; \tau_6 = \tau_{bc} \end{aligned}$$

and with

$$F_1 = \frac{1}{X_{at}} + \frac{1}{X_{ac}} ; F_{11} = -\frac{1}{X_{at}X_{ac}}$$

$$F_2 = \frac{1}{X_{bt}} + \frac{1}{X_{bc}} ; F_{22} = -\frac{1}{X_{bt}X_{bc}}$$

$$F_3 = \frac{1}{X_{ct}} + \frac{1}{X_{cc}} ; F_{33} = -\frac{1}{X_{ct}X_{cc}}$$

$$F_4, F_5, F_6 = 0$$

$$F_{44} = \frac{1}{S_{ab}^2} ; F_{55} = \frac{1}{S_{ac}^2} ; F_{66} = \frac{1}{S_{bc}^2}$$

where the following strength components need to be defined:

X_{at} = maximum allowable tensile stress in the material a-direction

X_{ac} = maximum allowable compressive stress in the material a-direction

X_{bt} = maximum allowable tensile stress in the material b-direction

X_{bc} = maximum allowable compressive stress in the material b-direction

X_{ct} = maximum allowable tensile stress in the material c-direction

X_{cc} = maximum allowable compressive stress in the material c-direction

S_{ab} = maximum allowable absolute shear stress in the material a-b plane

S_{ac} = maximum allowable absolute shear stress in the material a-c plane

S_{bc} = maximum allowable absolute shear stress in the material b-c plane

For plane stress conditions, the parameters X_{ct} , X_{cc} , S_{bc} , S_{ac} , F_{bc} and F_{ac} are not used.

- The interaction strengths F_{ij} with $i \neq j$ can be input, or calculated by ADINA according to Hoffman's convention:

$$F_{ij} = - \left(\frac{1}{X_{it}X_{ic}} + \frac{1}{X_{jt}X_{jc}} \right)$$

with $i \neq j$ and $i, j = a, b, c$

- Note that this failure criterion includes interactions between the failure modes, but that the mode of failure is not given by the analysis.
- The **Hashin failure criterion** applies especially to fibrous composites where the fiber and the matrix failure mechanisms are distinct. In ADINA, we assume that the fibers are predominantly aligned with the first material principal direction (a-direction), thus making the material transversely isotropic about the a-direction. Failure will occur when any of the following inequalities is not satisfied anymore:

$$\frac{\tau_a^2}{X_t^2} + \frac{(\tau_{ab}^2 + \tau_{ac}^2)}{S_{ab}^2} < 1 \quad (\text{if } \tau_a > 0) \quad (2.7-7)$$

$$\frac{\tau_a}{X_c} < 1 \quad (\text{if } \tau_a < 0) \quad (2.7-8)$$

$$\frac{(\tau_b + \tau_c)^2}{Y_t^2} + \frac{(\tau_{bc}^2 - \tau_b \tau_c)}{S_{tr}^2} + \frac{(\tau_{ab}^2 + \tau_{ac}^2)}{S_{ab}^2} < 1 \quad (\text{if } (\tau_b + \tau_c) > 0) \quad (2.7-9)$$

$$\frac{(\tau_b + \tau_c) \left[1 - \left(\frac{Y_c}{2S_{tr}} \right)^2 \right]}{Y_c} + \frac{(\tau_b + \tau_c)^2}{4S_{tr}^2} + \frac{(\tau_b^2 - \tau_b \tau_c)}{S_{tr}^2} + \frac{(\tau_{ab}^2 + \tau_{ac}^2)}{S_{ab}^2} < 1$$

(if $(\tau_b + \tau_c) < 0$)

(2.7-10)

with

- X_t = maximum allowable tensile stress in the material a-direction
- X_c = maximum allowable compressive stress in the

	material a-direction
Y_t	= maximum allowable tensile stress in the material b-direction
Y_c	= maximum allowable compressive stress in the material b-direction
S_{ab}	= maximum allowable absolute shear stress in the a-b plane (note that $S_{ab} = S_{ac}$)
S_{tr}	= maximum allowable absolute shear stress in the b-c plane

The above inequalities are usually referred to as: tensile fiber failure (2.7-7), compressive fiber failure (2.7-8), tensile matrix failure (2.7-9), compressive matrix failure (2.7-10).

- The **user-supplied failure criterion** allows for the description of a failure envelope using up to four quadratic failure relations of the form:

$$F_i \tau_i + F_{ij} \tau_i \tau_j = 1 \quad \text{if} \quad \sum_{i=1}^6 \alpha_i \tau_i > 0$$

with

$$\tau_1 = \tau_a \ ; \ \tau_2 = \tau_b \ ; \ \tau_3 = \tau_c$$

$$\tau_4 = \tau_{bc} \ ; \ \tau_5 = \tau_{ac} \ ; \ \tau_6 = \tau_{ab}$$

The material failure occurs when any one of the failure surfaces is reached and the corresponding stress condition is satisfied at the same time.

- This flexible and very general user-supplied failure model allows you to input any quadratic failure criteria with failure mode interactions.
- The following references contain additional information about composite failure criteria:

ref. Jones R.M., *Mechanics of Composite Materials*, McGraw Hill, 1975.

ref. Tsai S.W., "A Survey of Macroscopic Failure Criteria

for Composite Materials," *Journal of Reinforced Plastics and Composites*, Vol.3, pp.40-62, January 1984.

- ref. Hashin Z., "Analysis of Composite Materials – A Survey," *Journal of Applied Mechanics*, Transactions of the ASME, Vol. 50, pp. 481-505, September 1983.

2.7.9 Element output

You can request that ADINA either print and save stresses or forces.

Stresses: Each element outputs, at its integration points, the following information to the porthole file, based on the material model. This information is accessible in the AUI using the given variable names.

The results can be requested to be referred to the global (X,Y,Z) axes, or to the local (\bar{r}, \bar{s}, t) axes (see Fig. 2.7-6a), or to the midsurface $(\hat{r}, \hat{s}, \hat{t})$ axes (see Fig. 2.7-6b). For the large displacement/large strain formulations, both the local and midsurface axes are the $(\hat{r}, \hat{s}, \hat{t})$ axes, so the results in both the local and midsurface systems are identical. In addition, for orthotropic material models, the results referred to the material axes can be requested.

Elastic-isotropic, elastic-orthotropic: STRESS (XYZ) ,
STRAIN (XYZ) , FE_EFFECTIVE_STRESS

Elastic-isotropic with thermal effects: STRESS (XYZ) ,
STRAIN (XYZ) , FE_EFFECTIVE_STRESS,
THERMAL_STRAIN, ELEMENT_TEMPERATURE

Thermo-isotropic: STRESS (XYZ) , STRAIN (XYZ) ,
THERMAL_STRAIN (XYZ) , FE_EFFECTIVE_STRESS,
ELEMENT_TEMPERATURE

Plastic-bilinear, plastic-multilinear, without thermal effects:

PLASTIC_FLAG, STRESS(XYZ), STRAIN(XYZ),
PLASTIC_STRAIN(XYZ), FE_EFFECTIVE_STRESS,
YIELD_STRESS, ACCUM_EFF_PLASTIC_STRAIN

Plastic-bilinear, plastic-multilinear with thermal effects:

PLASTIC_FLAG, STRESS(XYZ), STRAIN(XYZ),
PLASTIC_STRAIN(XYZ), FE_EFFECTIVE_STRESS,
YIELD_STRESS, ACCUM_EFF_PLASTIC_STRAIN,
THERMAL_STRAIN(XYZ), ELEMENT_TEMPERATURE

Thermo-plastic, creep, plastic-creep, multilinear-plastic-creep,
creep-variable, plastic-creep-variable, multilinear-plastic-creep-

variable: PLASTIC_FLAG, NUMBER_OF_SUBINCREMENTS,
STRESS(XYZ), STRAIN(XYZ),
PLASTIC_STRAIN(XYZ), CREEP_STRAIN(XYZ),
THERMAL_STRAIN(XYZ), ELEMENT_TEMPERATURE,
ACCUM_EFF_PLASTIC_STRAIN,
FE_EFFECTIVE_STRESS, YIELD_STRESS,
EFFECTIVE_CREEP_STRAIN

Plastic-orthotropic without thermal effects: PLASTIC_FLAG,
STRESS(XYZ), STRAIN(XYZ),
PLASTIC_STRAIN(XYZ), HILL_EFFECTIVE_STRESS,
YIELD_STRESS, ACCUM_EFF_PLASTIC_STRAIN

Plastic-orthotropic with thermal effects: PLASTIC_FLAG,
STRESS(XYZ), STRAIN(XYZ),
PLASTIC_STRAIN(XYZ), HILL_EFFECTIVE_STRESS,
YIELD_STRESS, ACCUM_EFF_PLASTIC_STRAIN,
THERMAL_STRAIN(XYZ), ELEMENT_TEMPERATURE

Viscoelastic: STRESS(XYZ), STRAIN(XYZ),
THERMAL_STRAIN(XYZ), ELEMENT_TEMPERATURE

In the above lists,

when results are referred to the global (X,Y,Z) axes,
STRESS(XYZ) = STRESS-XX, STRESS-YY,
STRESS-ZZ, STRESS-XY, STRESS-XZ,
STRESS-YZ

when results are referred to the $(\bar{r}, \bar{s}, \bar{t})$ or $(\hat{r}, \hat{s}, \hat{t})$ axes,

```
STRESS(XYZ) = STRESS-RR, STRESS-SS,  
STRESS-TT, STRESS-RS, STRESS-RT,  
STRESS-ST
```

when results are referred to the material axes,

```
STRESS(XYZ) = STRESS-AA, STRESS-BB,  
STRESS-CC, STRESS-AB, STRESS-AC,  
STRESS-BC
```

with similar definitions for the other abbreviations used above.

See Section 13.1.1 for the definitions of those variables that are not self-explanatory.

- In addition, when a failure model is used, each element outputs, at each integration point, the following information to the porthole file, based on the failure criterion. This information is accessible in the AUI using the following variable names:

Maximum stress, maximum strain:

```
FAILURE_FLAG_TENSION-A,  
FAILURE_FLAG_COMPRESSION-A,  
FAILURE_FLAG_TENSION-B,  
FAILURE_FLAG_COMPRESSION-B,  
FAILURE_FLAG_TENSION-C,  
FAILURE_FLAG_COMPRESSION-C,  
FAILURE_FLAG_SHEAR-AB, FAILURE_FLAG_SHEAR-AC,  
FAILURE_FLAG_SHEAR-BC, FAILURE_FLAG
```

Tsai-Hill, tensor polynomial: FAILURE_FLAG,
FAILURE_CRITERION

Hashin: FAILURE_FLAG_TENS-FIBER,
FAILURE_FLAG_COMP-FIBER,
FAILURE_FLAG_TENS-MATRIX,
FAILURE_FLAG_COMP-MATRIX,
FAILURE_CRITERION_TENS-FIBER,
FAILURE_CRITERION_COMP-FIBER,
FAILURE_CRITERION_TENS-MATRIX,
FAILURE_CRITERION_COMP-MATRIX

User-supplied: FAILURE_FLAG_SURFACE-1,
FAILURE_FLAG_SURFACE-2,
FAILURE_FLAG_SURFACE-3,

```
FAILURE_FLAG_SURFACE-4, FAILURE_FLAG,  
FAILURE_CRITERION_SURFACE-1,  
FAILURE_CRITERION_SURFACE-2,  
FAILURE_CRITERION_SURFACE-3,  
FAILURE_CRITERION_SURFACE-4
```

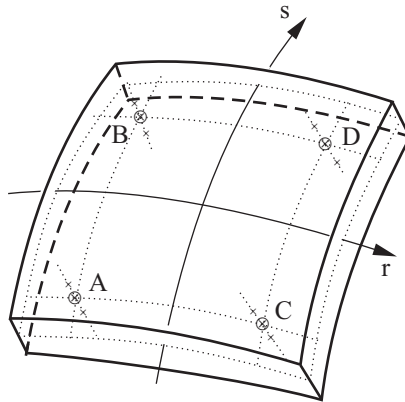
See Section 13.1.1 for the definitions of those variables that are not self-explanatory.

Section results: The computation of stress resultants, forces and moments, the computation of membrane strains and curvatures, and the position of the neutral axes can be requested in ADINA. The latter is especially useful for multilayered shell elements.

These results are given at the locations corresponding to the projections of the integration points on the shell element midsurface (see Fig. 2.7-16).

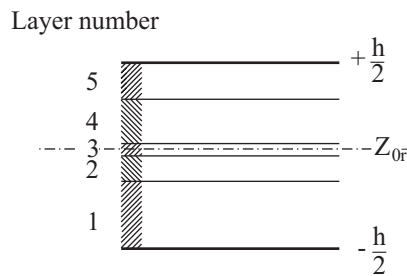
The position of the neutral axes is available only for multilayer shell elements.

Fig. 2.7-17 shows the convention used for positive forces and moments. Table 2.7-1 contains the formulas used to evaluate the stress and strain resultants and the neutral axes positions. They are always referred to the local Cartesian system (\bar{r}, \bar{s}, t) (see Fig. 2.7-16).



Point labels	INR	INS
A	1	1
B	1	2
C	2	1
D	2	2

a) Point labeling for section results
(2×2×2 Gauss integration)



b) Definition of the neutral axes

$$Z_{0\bar{r}} = \frac{\int_{-h/2}^{h/2} E_{\bar{r}} z dz}{\int_{-h/2}^{h/2} E_{\bar{r}} dz}$$

$$Z_{0\bar{s}} = \frac{\int_{-h/2}^{h/2} E_{\bar{s}} z dz}{\int_{-h/2}^{h/2} E_{\bar{s}} dz}$$

Figure 2.7-16: Conventions and definitions for section results output

Table 2.7-1: Section results output for the shell element

Stress resultants (forces and moments)	
$R_{\bar{r}\bar{r}} = \int_{-h/2}^{h/2} \sigma_{\bar{r}\bar{r}} dz$	$M_{\bar{r}\bar{r}} = \int_{-h/2}^{h/2} \sigma_{\bar{s}\bar{s}} z dz$
$R_{\bar{s}\bar{s}} = \int_{-h/2}^{h/2} \sigma_{\bar{s}\bar{s}} dz$	$M_{\bar{s}\bar{s}} = -\int_{-h/2}^{h/2} \sigma_{\bar{r}\bar{r}} z dz$
$R_{tt} = \int_{-h/2}^{h/2} \sigma_{tt} dz$	
$R_{\bar{r}\bar{s}} = \int_{-h/2}^{h/2} \sigma_{\bar{r}\bar{s}} dz$	$M_{\bar{r}\bar{s}} = \int_{-h/2}^{h/2} \sigma_{\bar{r}\bar{s}} z dz$
$R_{\bar{r}t} = \int_{-h/2}^{h/2} \sigma_{\bar{r}t} dz$	
$R_{\bar{s}t} = \int_{-h/2}^{h/2} \sigma_{\bar{s}t} dz$	
Membrane strains and curvatures	
$\varepsilon_{m\bar{r}\bar{r}} = \varepsilon_{l\bar{r}\bar{r}} - z_1 \frac{\varepsilon_{2\bar{r}\bar{r}} - \varepsilon_{l\bar{r}\bar{r}}}{(z_2 - z_1)}$	$\chi_{\bar{s}\bar{s}} = \frac{\varepsilon_{2\bar{r}\bar{r}} - \varepsilon_{l\bar{r}\bar{r}}}{(z_2 - z_1)}$
$\varepsilon_{m\bar{s}\bar{s}} = \varepsilon_{l\bar{s}\bar{s}} - z_1 \frac{\varepsilon_{2\bar{s}\bar{s}} - \varepsilon_{l\bar{s}\bar{s}}}{(z_2 - z_1)}$	$\chi_{\bar{r}\bar{r}} = \frac{\varepsilon_{2\bar{s}\bar{s}} - \varepsilon_{l\bar{s}\bar{s}}}{(z_2 - z_1)}$
$\varepsilon_{m\bar{r}\bar{s}} = \varepsilon_{l\bar{r}\bar{s}} - z_1 \frac{\varepsilon_{2\bar{r}\bar{s}} - \varepsilon_{l\bar{r}\bar{s}}}{(z_2 - z_1)}$	$\chi_{\bar{r}\bar{s}} = \frac{\varepsilon_{2\bar{r}\bar{s}} - \varepsilon_{l\bar{r}\bar{s}}}{(z_2 - z_1)}$

Indices 1 and 2 refer respectively to the lowest and highest integration points on a normal to the element midsurface.

Note that the section results are given with respect to the local Cartesian system of the shell elements.

The stress resultants do not satisfy equilibrium except in the limit as the mesh is refined.

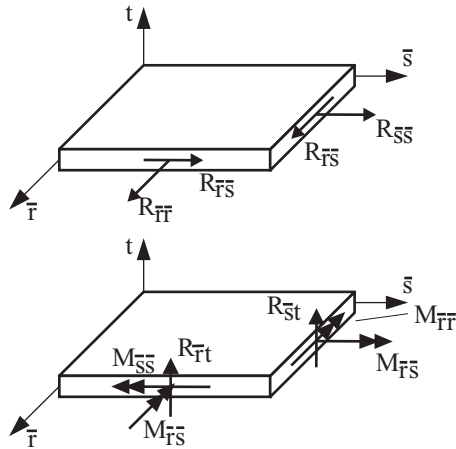


Figure 2.7-17: Nomenclature for shell stress resultants (positive)

The corresponding AUI variables are (for all material models):

MEMBRANE_FORCE-RB, MEMBRANE_FORCE-SB,
 MEMBRANE_FORCE-RBSB, SHEAR_FORCE-RB,
 SHEAR_FORCE-SB, BENDING_MOMENT-RB,
 BENDING_MOMENT-SB, BENDING_MOMENT-RBSB,
 MEMBRANE_STRAIN-RB, MEMBRANE_STRAIN-SB,
 MEMBRANE_STRAIN-RBSB, CURVATURE-RB,
 CURVATURE-SB, CURVATURE-RBSB,
 SHEAR_STRAIN-RB, SHEAR_STRAIN-SB,
 NEUTRAL_AXIS_POSITION-RB,
 NEUTRAL_AXIS_POSITION-SB.

See Section 13.1.1 for the definitions of those variables that are not self-explanatory.

*ref. KJB
 Example 5.11
 pp. 358-359*

Nodal forces: The nodal point force vector which corresponds to the element internal stresses can also be requested in ADINA. The procedure for the calculation of the force vector is given in ref. KJB.

The nodal forces are accessible in the AUI using the variable names NODAL_FORCE-X, NODAL_FORCE-Y, NODAL_FORCE-Z, NODAL_MOMENT-X, NODAL_MOMENT-Y, NODAL_MOMENT-Z.

2.7.10 Selection of elements for analysis of thin and thick shells

- The following types of shell elements are available:

	Number of nodes	Single layer with mid-surface nodes only	Multilayer or top & bottom
Quadrilateral elements	4-node	MITC4 / MITC4+	MITC4
	8-node	MITC8	MITC8
	9-node	MITC9 / MITC3+	DISP9
	16-node	MITC16	DISP16
Triangular elements	3-node	MITC3	MITC4 Collapsed
	6-node	MITC6	DISP7 Collapsed
	9-node	DISP10 Collapsed	DISP10 Collapsed

The types of triangular elements are illustrated in Fig. 2.7-18.

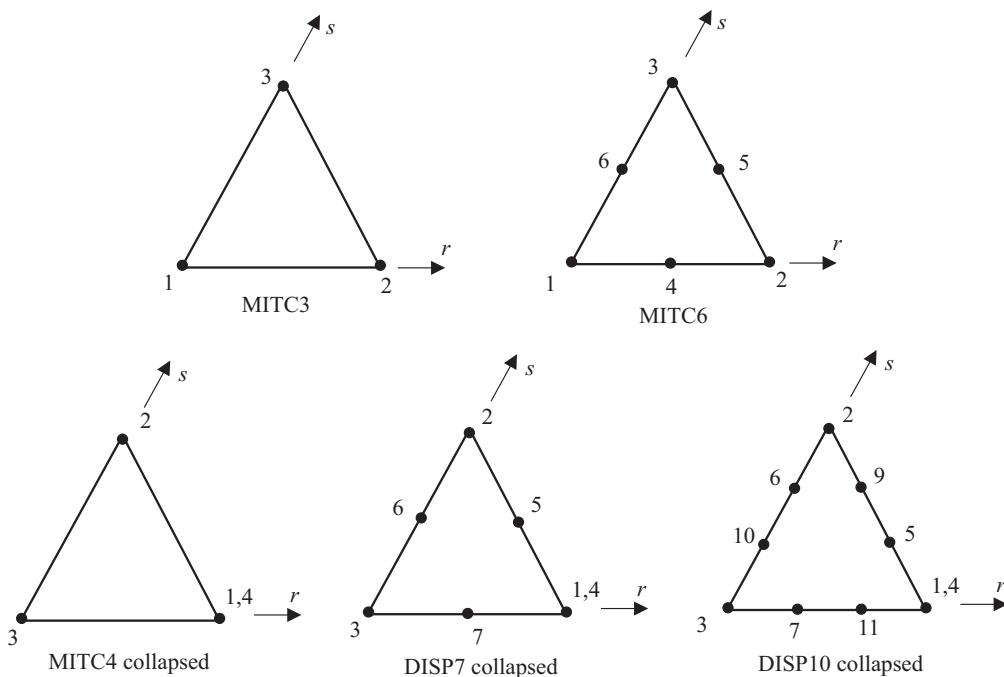


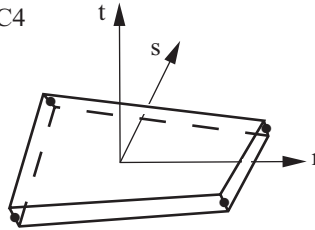
Figure 2.7-18. Types of triangular elements

- The most effective element for analysis of general shells is usually the 4-node element, shown in Fig. 2.7-19(a). This element does not lock and has a high predictive capability and hence can be

used for effective analysis of thin and thick shells. See references:

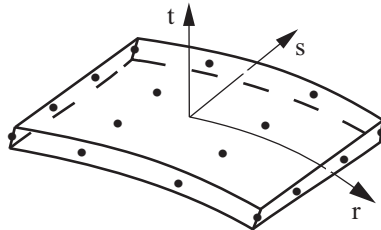
- ref. Dvorkin, E. and Bathe, K.J., "A Continuum Mechanics Based Four-Node Shell Element for General Nonlinear Analysis," *Engineering Computations*, Vol. 1, pp. 77-88, 1984.

Recommended element: MITC4

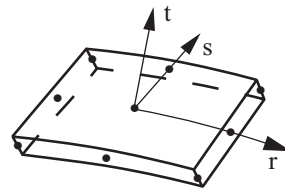


(a) 4-node shell element for thick and thin shells.

Sometimes also useful elements: MITC16 and MITC9



MITC16



MITC9

(b) 16-node cubic shell element for thick and thin shells.
9-node shell element for thick and thin shells.

Figure 2.7-19: Recommended elements for analysis of any shell

- ref. Bathe, K.J. and Dvorkin, E., "A Four-Node Plate Bending Element Based on Reissner/Mindlin Plate Theory and a Mixed Interpolation," *Int. J. Num. Meth. in Eng.*, Vol. 21, pp. 367-383, 1985.
- ref. Bathe, K.J. and Dvorkin, E., "A Formulation of General

Shell Elements – The Use of Mixed Interpolation of Tensorial Components," *Int. J. Num. Meth. in Eng.*, Vol. 22 pp. 697-722, 1986.

- ref. Bucalem, M.L and Bathe, K.J., "Higher-Order MITC General Shell Elements," *Int. J. for Num. Meth in Engng*, Vol. 36, pp. 3729-3754, 1993.
- ref. Bathe, K.J., Iosilevich, A. and Chapelle, D., "An Evaluation of the MITC Shell Elements," *Comp. Struct*, Vol. 75, pp.1-30, 2000.
- ref. Lee, P.H. and Bathe, K.J., "Development of MITC isotropic triangular shell finite elements", *Comp. Struct*, Vol.82, 945-962, 2004.

In linear analysis it can be effective to use the 9-node shell element (the MITC9 element) or the 16-node shell element (the MITC16 element) (see Fig. 2.7-19(b)). However, the use of these elements can be costly.

ref. KJB
pp. 403-408

- The phenomenon of an element being much too stiff is in the literature referred to as element locking. In essence, the phenomenon arises because the interpolation functions used for an element are not able to represent zero (or very small) shearing or membrane strains. If the element cannot represent zero shearing strains, but the physical situation corresponds to zero (or very small) shearing strains, then the element becomes very stiff as its thickness over length ratio decreases.

The MITC3, MITC4, MITC6, MITC8, MITC9 and MITC16 elements are implemented to overcome the locking problem in the shell element. When you choose to generate the 4-node shell element (through the AUI), the MITC4 elements are created for the quadrilateral elements and MITC3 elements are used for the triangular elements. When you choose to generate the 8-node shell element, the MITC8 elements are created for the quadrilateral elements and the MITC6 elements are created for the triangular elements. When you choose to generate the 9-node shell element, the MITC9 elements are created for the quadrilateral elements and the MITC6 elements are used for the triangular elements. When you choose to generate the 16-node shell element, the MITC16

elements are created for the quadrilateral elements and DISP10 collapsed triangular elements are used for the triangular elements. The MITC8 element is not as effective as the other MITC elements, and its use is not recommended in general.

- In order to arrive at an appropriate element idealization of a thin shell, it may be effective to consider the behavior of a single element in modeling a typical part of the shell. As an example, if a shell of thickness h and principal radii of curvatures R_1 and R_2 is to be analyzed, a single element of this thickness and these radii and supported as a cantilever could be subjected to different simple stress states. The behavior of the single element when subjected to the simple stress states (e.g., constant bending moments) tells what size of element, and hence element idealization, can be used to solve the actual shell problem.
- For the analysis of thick shells, the elements depicted in Fig. 2.7-20 can be used as well. If necessary, these elements can also be employed together with the elements of Fig. 2.7-19 in the analysis of a thin shell, but then only a few elements should be employed to model a special region, where necessary, such as a transition region, a cut-out, and a triangular region. Also, the elements should be small enough in size.

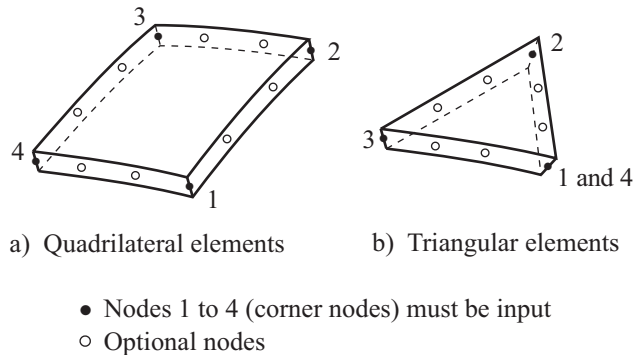


Figure 2.7-20: Elements for analysis of thick shells

- For cases where the ratio thickness/radius of curvature is large (in the original configuration or in the deformed configuration in large deformation analysis) it is best to use 5 dofs at each shell

node. Then all elements will represent the hyperbolic stress distributions through the element thickness. If 6 dofs per node are used, then only the higher order elements will represent the hyperbolic stress distribution because the nodal director vectors are constructed from the geometry of the elements.

- It is recommended that, as often as possible, only the 4-node, 9-node and 16-node elements are employed for the quadrilateral elements, 3-node and 6-node elements are employed for the triangular elements.
- Note that the elements in Fig. 2.7-20 can also be used as transition elements by assigning top and bottom surface nodes instead of a midsurface node.
- The shell elements can be used in conjunction with the beam-bolt elements (Section 11.17) to model plates connected by bolts.
- Geometrically nonlinear incompatible modes elements with large aspect ratio should not be used, because spurious modes may be present in the finite element solution.

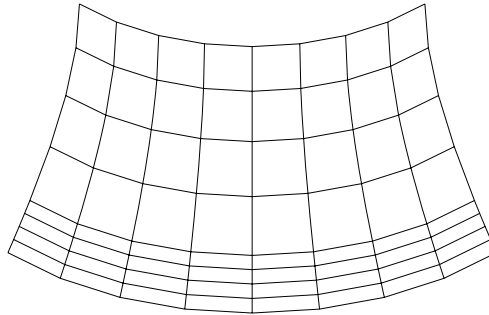
2.7.10.1 MITC+ formulation

- The 3-node and 4-node lower-order elements are in general more effective than the higher-order 6-node, 8-node, 9-node and 16-node elements.
- However, when the shell structure is very thin (thickness to overall dimension ratio $t/L=1/1000$ or lower), and distorted meshes are used, locking effects can appear for 3-node and 4-node MITC elements.
- To improve the accuracy of 3-node and 4-node MITC element solutions, the MITC+ formulation can be used. Shell elements with the MITC+ formulation are particularly effective when the finite element model contains out-of-plane distorted elements. This can be the case in the initial geometry or after large deformations have taken place. Examples are shown in Fig 2.7-21, Fig 2.7-22 and Fig. 2.7-23.

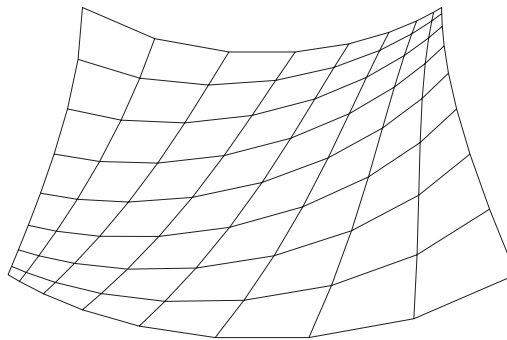
For the mesh in Fig. 2.7-21 (a) the MITC4 element mesh gives optimal solutions but when the elements are distorted as shown in Fig. 2.7-21(b) the solution accuracy deteriorates. Then the MITC4+ element is more powerful, indeed it then gives still excellent results.

- The MITC+ formulation can be used in static, implicit dynamic, explicit dynamic or frequency analysis.
- The formulation of the shell elements is specified in the element group. When EGROUP SHELL FORMULATION=MITC, the original MITC formulation is chosen and when EGROUP SHELL FORMULATION=MITC-PLUS, the MITC+ formulation is chosen.
- The MITC+ formulation is described in the following references:

- ref. Y. Lee, P.S. Lee & K.J. Bathe, The MITC3+ shell element and its performance, *Computers & Structures*, **138**, 2014.
- ref. Y. Ko, P.S. Lee & K.J. Bathe, A new MITC4+ shell element, *Computers & Structures*, **182**, 2017.

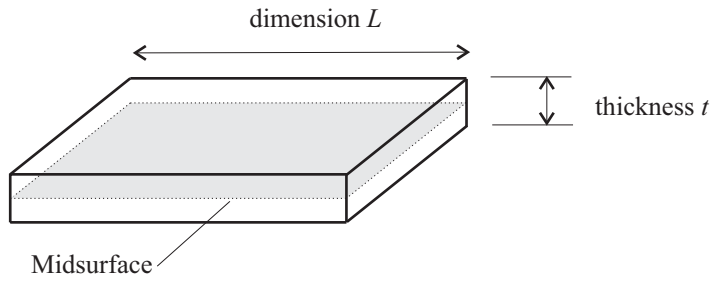


(a)

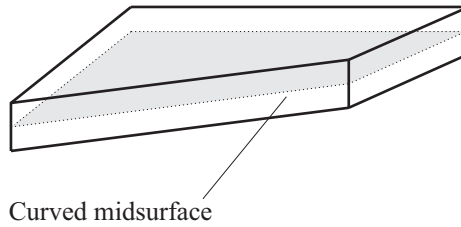


(b)

Figure 2.7-21: Shells with doubly curved meshes

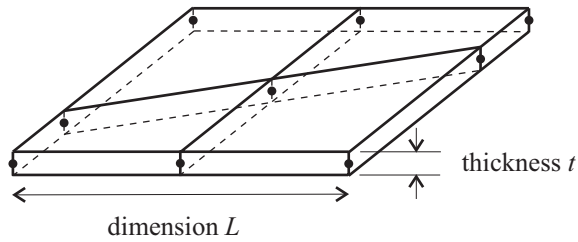


a) Undeformed configuration

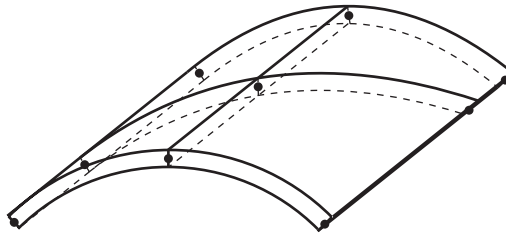


b) Deformed configuration

Figure 2.7-22: Shell analysis using curved mesh



a) Undeformed configuration



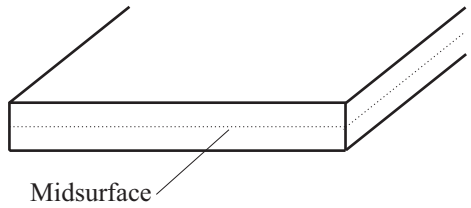
b) Deformed configuration

Figure 2.7-23: Pure bending modeled with distorted mesh

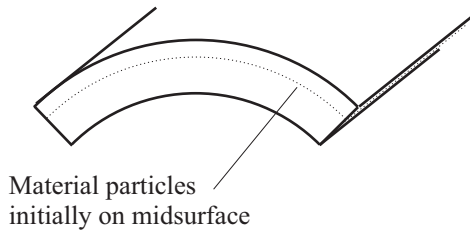
2.7.11 3D-shell element

Overview

- One characteristic of the shell elements described earlier is that the change in thickness of the element is not explicitly calculated from the element degrees of freedom. This is because the zero stress through the shell thickness assumption is used in the material descriptions.
- However, in large strain analysis, the change in thickness can become important. For example, during out-of-plane bending, the material in compression thickens, and the material in tension thins (for nonzero Poisson's ratio). Hence the midsurface is no longer exactly halfway between the top and bottom surfaces. This effect is shown in Fig 2.7-24.



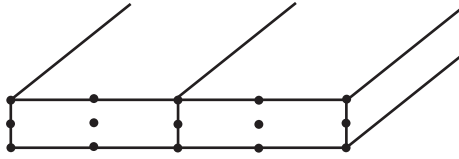
a) Undeformed configuration



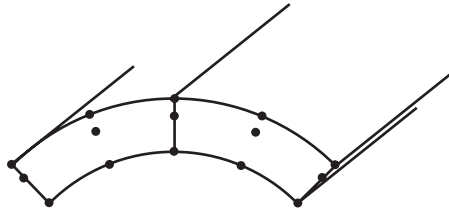
b) Deformed configuration

Figure 2.7-24: Kinematics of pure bending

- This change in thickness is naturally modeled when quadratic 3D solid elements are used, because the nodes on the top and bottom surfaces can move relative to the nodes on the midsurface, as shown in Fig 2.7-25. However 3D solid elements tend to lock when they are very thin, so that they are unsuitable for bending analysis of thin structures.



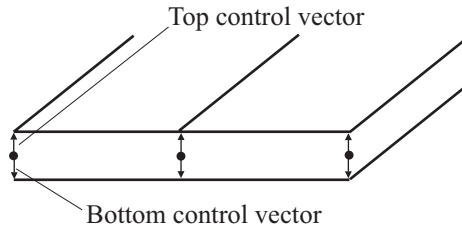
a) Undeformed configuration



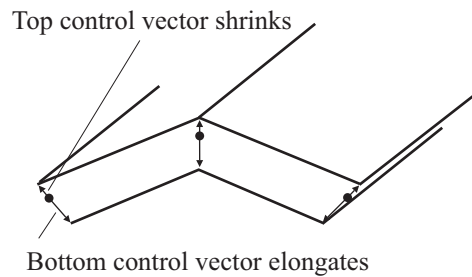
b) Deformed configuration

Figure 2.7-25: Pure bending modeled with quadratic 3D elements

- In the 3D-shell element, the change in thickness of the element is modeled using control vectors, as shown for the case of pure bending in Fig 2.7-26. The motion of the control vectors is controlled by element degrees of freedom at the shell midsurface, as discussed in more detail below.



a) Undeformed configuration



b) Deformed configuration

Figure 2.7-26: Pure bending modeled with 3D-shell elements

- Because the change in thickness is explicitly calculated from element degrees of freedom, the assumption of zero stress through the shell thickness is not used in the 3D-shell element.
- In addition, the 3D-shell elements use MITC tying rules to relieve shear locking. Therefore these elements are suitable for out-of-plane bending analysis of thin structures, even for large bending strains.
- The 3D-shell elements can be used with 3 or 4 nodes. The 4-node element is recommended for general use.
- The 3D-shell elements can be used in static, implicit dynamic, explicit dynamic or frequency analysis.

- The 3D-shell elements cannot be used in the following types of analysis: cyclic symmetry or periodic symmetry, thermo-mechanical-coupling analysis, in which the structural and thermal models are specified by two input files.
- The following features available for other shell elements are not available for the 3D-shell element: 6-, 8-, 9-, 16-node elements, composite (multilayer) shells, transition elements, anisotropic failure criteria. When section results are requested, the membrane strains and curvatures are not available.
- When using contact groups with true offsets (CGROUP ... OFFSET-TYPE=TRUE), the 3D-shell elements contribute to the offsets. However, the offset at a 3D-shell element node is computed as half the current shell thickness at the node, and this offset is used regardless of whether the contact occurs on the shell top surface or on the shell bottom surface.
- The 3D-shell is more fully described in the following reference:

ref. T. Sussman & K.J. Bathe, 3D-shell elements for structures in large strains, *Computers & Structures*, **122**, 2-12, 2013.

Kinematics and degrees of freedom of the 3D-shell element

The kinematics of the 3D-shell element are similar to the kinematics of the quadratic 3D element. The correspondence between the quadratic 3D element and the 3D-shell element is shown in Fig 2.7-27. Fig 2.7-27(a) shows a cross-section of a shell structure that has undergone large deformations. Three material particles are labeled: A at the midsurface, B at the top surface and C at the bottom surface. In both elements, the position of material particle A is given by the position of a node at A. Now consider the material particles B and C. In the quadratic 3D element, the positions of material particles B and C are given by the positions of nodes at B and C. In the 3D-shell element, the positions of material particles B and C are given by the position of the node at A, and also two control vectors. In both elements, the line of material particles between points B and C is specified by the positions of material particles A, B, C.

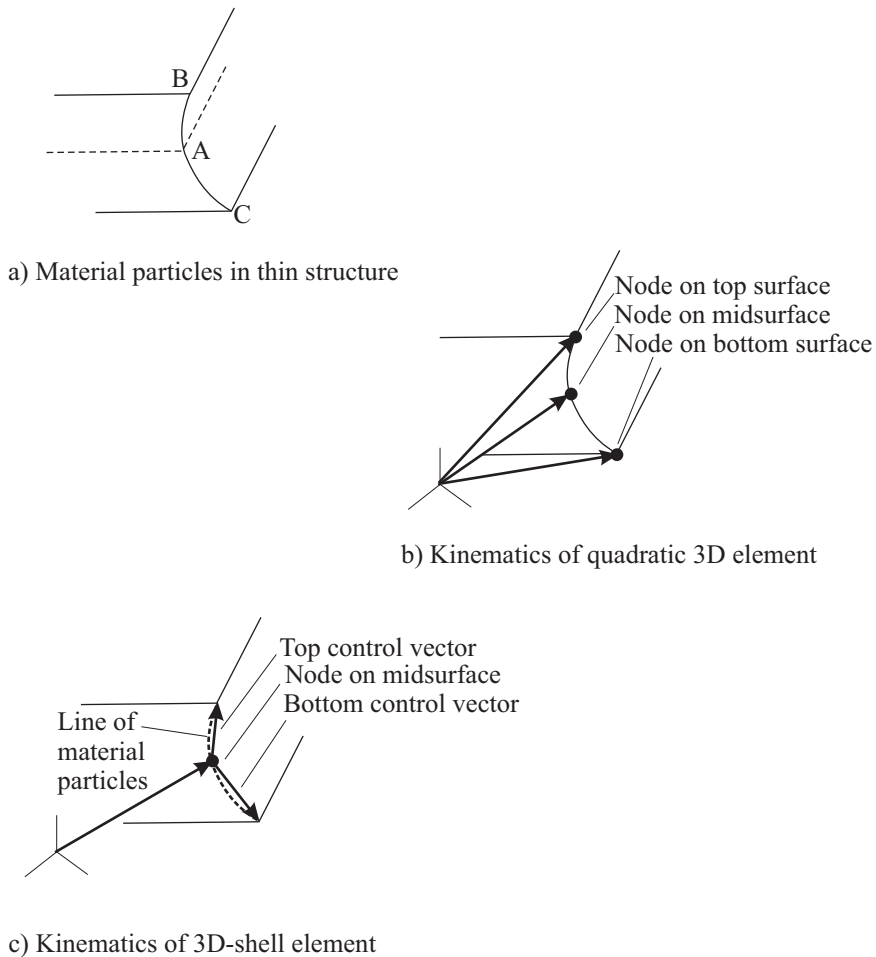


Figure 27.7-27: Correspondence between quadratic 3D element and 3D-shell element

Fig. 2.7-28 shows the corner of a 3D-shell element, with its top and bottom surfaces described by control vectors. Initially the control vectors are equal and opposite. During deformations, the control vectors can evolve independently. Thus in the deformed configuration, the control vectors are in general not equal and opposite, as shown in Fig 2.7-28(b).

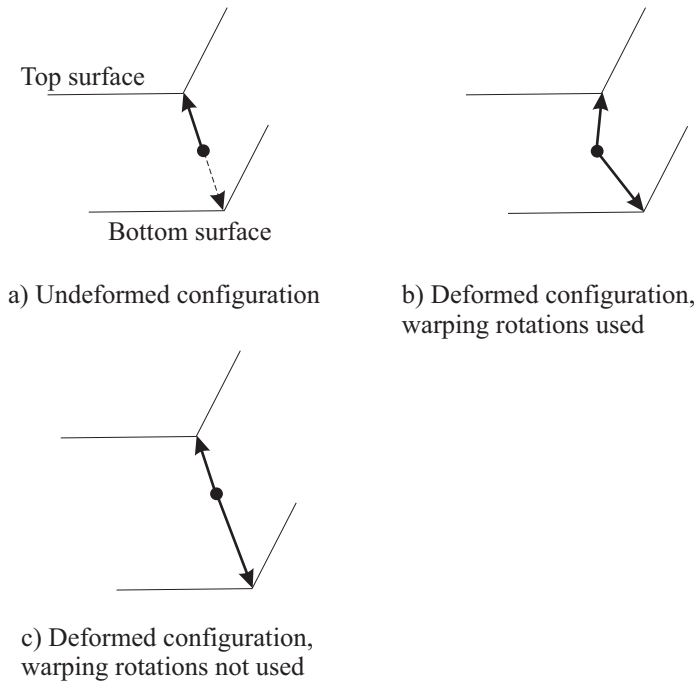


Figure 2.7-28: Control vectors at 3D-shell element node

The control vector motions are governed by element degrees of freedom. For ease of use of the element, the element degrees of freedom include the same degrees of freedom as for the usual shell elements:

x, y, z translations
 α , β rotations (5 DOF node), or θ_x , θ_y , θ_z rotations (6 DOF node)

and extra degrees of freedom:

constant and linear thickness incremental strains
 $\tilde{\alpha}$, $\tilde{\beta}$ rotations (5 DOF node), or $\tilde{\theta}_x$, $\tilde{\theta}_y$, $\tilde{\theta}_z$ rotations (6 DOF node)

The elongations of the control vectors are governed by the constant and linear thickness incremental strains. The directions of the

control vectors relative to each other are governed by the additional rotations, which are termed "warping rotations". (The term "warping rotation" is used because the additional rotational degrees of freedom cause the material particles initially on a straight line to warp into a curved quadratic line.)

Typically the 3D-shell element is used without warping rotations, then the extra degrees of freedom are the constant and linear thickness strains. In this case the control vectors always point in opposite directions, but in general have different lengths after deformations. And material particles that were initially on a straight line remain on a straight line after deformations. This situation is shown in Fig 2.7-28(c).

All of the considerations for selection of 5 and 6 DOF nodes discussed in Section 2.7.3 directly apply to the 3D-shell element nodes.

It is not allowed to prescribe or fix any of the extra degrees of freedom. If any of the usual rotations are prescribed or fixed, then the AUI automatically fixes the corresponding warping rotations.

Material models and formulations for the 3D-shell element

Material models for the usual shell elements are developed using the assumption of zero stress through the shell thickness. Hence these models do not directly apply to the 3D-shell element.

The following material models are implemented for the 3D-shell element: **elastic-isotropic, plastic-orthotropic, plastic-cyclic, Mooney-Rivlin, Ogden, Arruda-Boyce, hyper-foam, Sussman-Bathe, eight-chain, three-network.**

Small displacement/small strain, large displacement/small strain or large displacement/large strain formulations can be used with these material models, as described under the material models descriptions.

Material model notes

The thermal strains are assumed to be small in all material models,

including the rubber-like material models.

When stresses are output in the shell local coordinate system, the coordinate system is the $\hat{r}, \hat{s}, \hat{t}$ system shown in Figure 2.7-6(b).

No shear correction factor is used in the 3D-shell elements.

Mixed u/p formulation

The mixed u/p formulation should be used when the material is almost incompressible, to avoid volumetric locking. When the u/p formulation is used, by default the assumed pressure field is $p = p_0 + p_1 t$ where p_0 and p_1 are the pressure degrees of freedom, and t is the isoparametric coordinate through the shell thickness. Note that it is necessary to allow a linear variation of pressure through the shell thickness in order to model out-of-plane bending.

It is also allowed to select the number of pressure degrees of freedom in the plane of the shell, and the number of pressure degrees of freedom used through the shell thickness. The case $p = p_0 + p_1 t$ corresponds to one pressure degree of freedom in the plane of the shell and two pressure degrees of freedom through the shell thickness.

Incompatible modes can be used in the 4-node 3D-shell element. However incompatible modes and the u/p formulation cannot be used together.

The tables at the end of this section show under what conditions the incompatible modes and u/p formulations are used. Notice that these conditions are material dependent.

Element group input

3D-shell elements are placed in a shell element group (EGROUP SHELL), with the following additional command parameters: SHELLTYPE, WARPING_ROTATIONS, NELPRS, NELPT, TYING. SHELLTYPE=3DSHELL activates the 3D-shell elements.

Warping rotations are specified using parameter WARPING_ROTATIONS. Warping rotations are not used by default.

The u/p formulation is specified using parameters NELPRS and NELPT. NELPRS is the number of pressure degrees of freedom in the plane of the shell, and NELPT is the number of pressure degrees of freedom through the shell thickness. These parameters have the possible values 0 (no u/p), >0, (u/p) or AUTOMATIC (u/p depending upon the material), see tables at the end of this section.

Tying is specified using parameter TYING. TYING is used by default.

The interaction between incompatible modes, u/p formulation and the material models is given in the following tables. The default behavior for each material is given in the first row of each table.

Materials that do not allow the u/p formulation: Hyper-foam material

KINEMATICS INCOMPATIBLE- MODES	EGROUP INCOMPATIBLE- MODES	EGROUP NELPRS / NELPT	Result
AUTOMATIC or NO	DEFAULT	AUTOMATIC	No u/p or incompatible modes
YES	DEFAULT	AUTOMATIC	Incompatible modes used
	NO	AUTOMATIC	No u/p or incompatible modes
	YES	AUTOMATIC	Incompatible modes used
AUTOMATIC or NO	DEFAULT	0 / 0	No u/p or incompatible modes
YES	DEFAULT	0 / 0	Incompatible modes used
	NO	0 / 0	No u/p or incompatible modes
	YES	0 / 0	Incompatible modes used
AUTOMATIC or NO	DEFAULT	A / B	Error
YES	DEFAULT	A / B	Error
	NO	A / B	Error
	YES	A / B	Error

Materials that allow the u/p formulation, but for which the u/p formulation is not the default: Elastic material

KINEMATICS INCOMPATIBLE- MODES	EGROUP INCOMPATIBLE- MODES	EGROUP NELPRS / NELPT	Result
AUTOMATIC or NO	DEFAULT	AUTOMATIC	No u/p or incompatible modes
YES	DEFAULT	AUTOMATIC	Incompatible modes used
	NO	AUTOMATIC	No u/p or incompatible modes
	YES	AUTOMATIC	Incompatible modes used
AUTOMATIC or NO	DEFAULT	0 / 0	No u/p or incompatible modes
YES	DEFAULT	0 / 0	Incompatible modes used
	NO	0 / 0	No u/p or incompatible modes
	YES	0 / 0	Incompatible modes used
AUTOMATIC or NO	DEFAULT	A / B	u/p used with A*B pressure dofs
YES	DEFAULT	A / B	Error
	NO	A / B	u/p used with A*B pressure dofs
	YES	A / B	Error

Materials that allow the u/p formulation, and for which the u/p formulation is the default: Plastic-cyclic, plastic-orthotropic, Ogden, Mooney-Rivlin, Arruda-Boyce, Sussman-Bathe materials, eight-chain, three-network model

KINEMATICS INCOMPATIBLE- MODES	EGROUP INCOMPATIBLE- MODES	EGROUP NELPRS / NELPT	Result
AUTOMATIC or NO	DEFAULT	AUTOMATIC	Automatic u/p
YES	DEFAULT	AUTOMATIC	Incompatible modes used
	NO	AUTOMATIC	Automatic u/p
	YES	AUTOMATIC	Incompatible modes used
AUTOMATIC or NO	DEFAULT	0 / 0	No u/p or incompatible modes
YES	DEFAULT	0 / 0	Incompatible modes used
	NO	0 / 0	No u/p or incompatible modes
	YES	0 / 0	Incompatible modes used
AUTOMATIC or NO	DEFAULT	A / B	u/p used with A*B pressure dofs
YES	DEFAULT	A / B	Error
	NO	A / B	u/p used with A*B pressure dofs
	YES	A / B	Error

Automatic u/p: u/p used with 1 / 2 pressure dofs, i.e. $p = p_0 + p_1 t$

2.8 Pipe elements

- Two types of pipe element are available: **pipe-beam element** and **pipe-shell element**. The program automatically selects one of these element types depending on the setting used for the pipe element. If no ovalization or warping of the cross-section are present, the pipe-beam element is used. If ovalization or warping of the cross-section are present, the pipe-shell element is used. Only

4-node elements are supported. Figure 2.8-1 shows the conventions used.

- Assemblages of pipe elements can be subjected to internal pressure, see Section 2.8.2.
- The pipe-beam element has 6 displacement degrees of freedom per node (3 translations and 3 rotations, see Fig. 2.8-1).

ref. KJB
Sections 5.4.1,
6.5.1

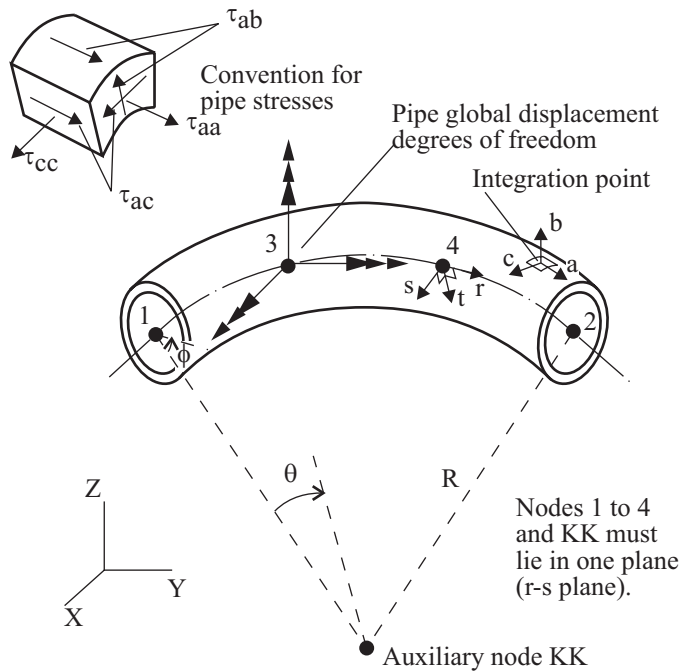


Figure 2.8-1: Pipe element configuration and displacement degrees of freedom and stress convention

- The pipe-shell element has, in addition to the 6 pipe-beam degrees of freedom, 3 or 6 ovalization and 3 or 6 warping degrees

of freedom per node allowing the cross-section to ovalize and warp. Ovalization refers to the deformation of the pipe-skin in the cross-sectional plane, and warping refers to the deformation of the pipe-skin in the direction normal to the pipe cross-section. The additional degrees of freedom are allocated as follows: 3 ovalization and 3 warping degrees of freedom for in-plane bending deformation; 3 ovalization and 3 warping degrees of freedom for out-of-plane bending deformation; 6 ovalization and 6 warping degrees of freedom for combined in-plane and out-of-plane bending. The user determines whether to activate one or both of in-plane or out-of-plane warping/ovalization.

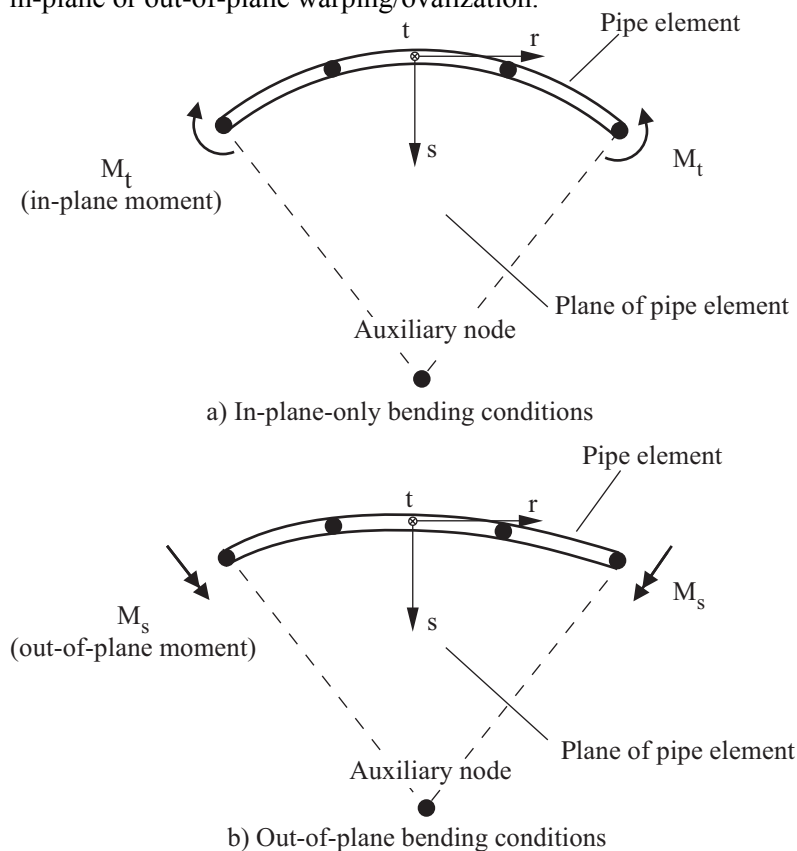


Figure 2.8-2: Loading conditions for the pipe element

- Note that the in-plane bending in this terminology corresponds to the bending in the r - s plane and the out-of-plane bending

corresponds to the bending in the r-t plane (see Fig. 2.8-1 and 2.8-2).

- The ovalization degrees of freedom are based on the von Karman ovalization modes, with the following circumferential displacement:

$$w_{\xi} = \sum_{m=1}^{N_c} c_m \sin 2m\phi + \sum_{m=1}^{N_d} d_m \cos 2m\phi$$

where c_m and d_m are generalized ovalization displacements which are interpolated from their nodal values using polynomial shape functions. The c_m terms correspond to in-plane bending and the d_m terms correspond to out-of-plane bending. N_c and N_d are the number of von Karman modes used in the analysis for in-plane and out-of-plane bending actions ($N_c = N_d = 3$ is always used in ADINA).

- Similarly, warping of the cross-section is described by a longitudinal displacement field analogous to the interpolation for the ovalization displacement

$$w_{\eta} = \sum_{m=1}^{N_c} p_m \cos 2m\phi + \sum_{m=1}^{N_d} q_m \sin 2m\phi$$

where p_m and q_m are generalized ovalization displacements which are interpolated from their nodal values using polynomial shape functions. The p_m terms correspond to in-plane bending and the q_m terms correspond to out-of-plane bending.

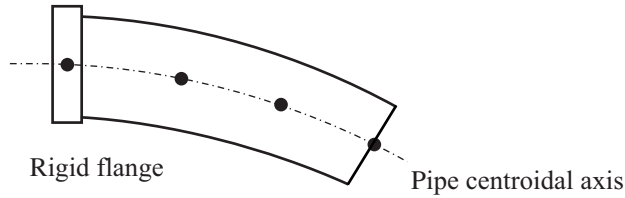
The formulation of the pipe-shell element (ovalization/warping included) is described in more detail in the following references:

- ref. K.J. Bathe and C.A. Almeida, "A Simple and Effective Pipe Elbow Element – Linear Analysis," *J. Appl. Mech., Transactions of the ASME*, Vol. 47, No. 1, pp. 93-100, 1980.
- ref. K.J. Bathe, C.A. Almeida and L.W. Ho, "A Simple and Effective Pipe Elbow Element – Some Nonlinear Capabilities," *Comp. & Structures*, Vol. 17, No. 5/6, pp.

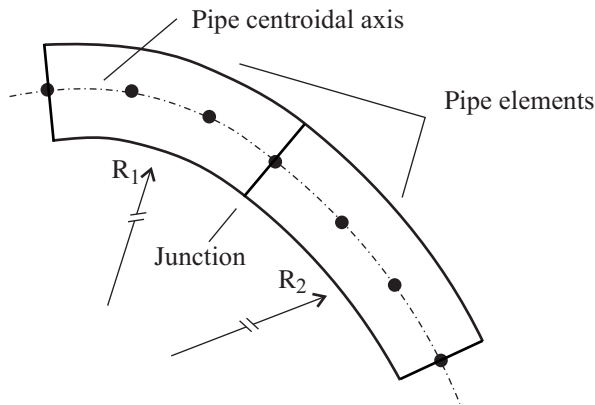
659-669, 1983.

- The pipe-shell formulation also takes into account the effects of stiffening due to internal pressure, and the effects of interactions between pipes and flanges, and pipes with different radii, see Fig. 2.8-3. The formulations are described in the following references:

- ref. K.J. Bathe and C.A. Almeida, "A Simple and Effective Pipe Elbow Element – Interaction Effects," *J. Appl. Mech., Transactions of The ASME*, Vol. 49, pp. 165-171, 1982.
- ref. K.J. Bathe and C.A. Almeida, "A Simple and Effective Pipe Elbow Element – Pressure Stiffening Effects," *J. Appl. Mech., Transaction of the ASME*, Vol. 49, pp. 914-916, 1982.



a) Pipe connected to a rigid flange



b) Pipes of different radii joined together

Figure 2.8-3: Interaction effects with the pipe element

- Whereas the pipe-beam element models the usual beam strains only, the pipe-shell element models in addition ovalization and warping effects by including the following strains (see Fig. 2.8-1 for notation):

- The ovalization is included by the von Karman ovalization modes with

$$w_{\zeta} = \frac{\partial w_{\xi}}{\partial \phi}$$

and

$$e_{aa}^{ov} = \frac{w_{\xi} \sin \phi - w_{\zeta} \cos \phi}{R - a \cos \phi} - \left(\frac{1}{(R - a \cos \phi)^2} \frac{\partial^2 w_{\zeta}}{\partial \theta^2} \right) \zeta$$

$$\gamma_{ab}^{ov} = 0 \quad (2.8-1)$$

$$\gamma_{ac}^{ov} = \frac{1}{R - a \cos \phi} \frac{\partial w_{\zeta}}{\partial \theta}$$

$$e_{cc}^{ov} = -\frac{1}{a^2} \left(w_{\zeta} + \frac{\partial^2 w_{\zeta}}{\partial \phi^2} \right) \zeta$$

► The warping is included using shell theory as

$$e_{aa}^w = \left(1 - \frac{\cos \phi}{R - a \cos \phi} \zeta \right) \left(\frac{1}{R - a \cos \phi} \right) \frac{\partial w_{\eta}}{\partial \theta}$$

$$\gamma_{ab}^w = 0$$

$$\gamma_{ac}^w = \frac{1}{a} \frac{\partial w_{\eta}}{\partial \phi} - \frac{w_{\eta} \sin \phi}{R - a \cos \phi} \quad (2.8-2)$$

$$e_{cc}^w = 0$$

Here, a is the pipe mean radius, R is the bend radius and θ and ϕ are the angles defined in Fig. 2.8-1.

In the formulation of the element the displacements w_{ζ} for the ovalization in Eq. 2.8-1 and w_{η} for the warping in Eq. 2.8-2 are interpolated by sine and cosine functions associated with the ovalization and warping degrees of freedom.

- For the pipe-shell element, continuity of ovalization and warping generalized displacements is enforced by assigning the same degrees of freedom to the node which the adjacent elements are attached. However, a restriction in ADINA is that the continuity

of the derivative $\partial w_{\zeta} / \partial \eta$ is only enforced between elements of the same element group. Hence, elements which model a region of significant longitudinal variation of the ovalization should be placed into one element group.

- A further restriction when using the pipe-shell element in ADINA is that for continuity in the ovalization and warping displacements, adjacent elements should lie in one plane. This is because the ovalization and warping displacements are measured using the angle ϕ for the s-axis of each element. In practice, adjacent bend elements lying in different planes are usually joined by straight sections (or a rigid flange) and these cases can be accurately modeled with the pipe-shell elements, because all ovalization and warping displacements are approximately zero in the middle of the straight section and at the rigid flange.
- Note that without warping/ovalization, the pipe-beam and pipe-shell elements give the same stress response. Hence, the ovalization and warping of the pipe-shell element can be thought of as additional kinematic modes to the pipe-beam response which result in an appropriate reduction of stiffness of pipe bends.
- For the pipe-shell element, the following use of the ovalization/warping degrees of freedom is recommended:
 - ▶ Activate the in-plane ovalization/warping degrees of freedom if the pipe elements undergo only in-plane deformations, see Fig. 2.8-2(a).
 - ▶ Activate the out-of-plane ovalization/warping degrees of freedom if the pipe elements undergo only out-of-plane deformations, see Fig. 2.8-2(b).
 - ▶ Activate all ovalization/warping degrees of freedom if the pipe elements undergo general three-dimensional deformations.
- In order to model the presence of a flange at a pipe node (of a pipe-shell element), all ovalization/warping degrees of freedom should be deleted at the node. The deletion of all nodal ovalization/warping degrees of freedom implies that the pipe section at that node remains circular. In addition, the option of

enforcing a zero pipe-skin slope must be employed to model the flange (see Fig. 2.8-4).

- The symmetry condition, also shown in Fig. 2.8-4, implies zero warping but not necessarily zero ovalization.
- For a pipe elbow, a nondimensional geometric factor λ is defined as

$$\lambda = \frac{R\delta}{a^2\sqrt{1-\nu^2}}$$

where R = pipe bend radius, δ = pipe wall thickness, a = pipe cross-section mean radius, ν = Poisson's ratio.

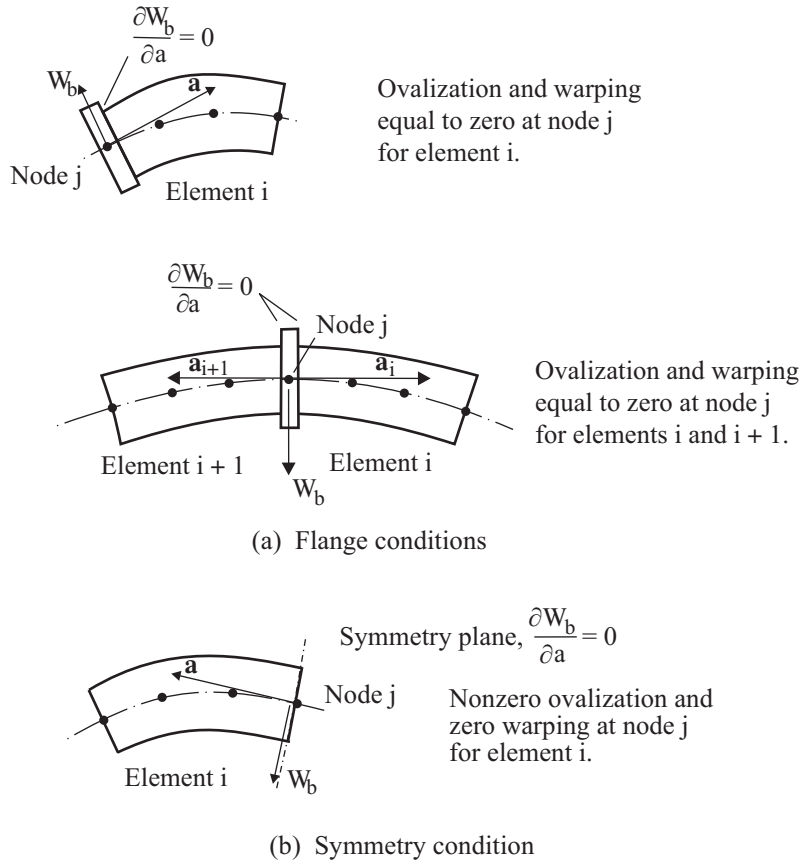


Figure 2.8-4: Zero pipe-skin slope conditions

With a decreasing value of λ , the ovalization/warping effects of the cross-section become more pronounced. Note that for a straight pipe element, $R \rightarrow \infty$ and thus the cross-section does not ovalize/warp except if connected to another pipe element with a finite bend radius.

2.8.1 Material models and formulations

- The pipe elements can be used with the following material models: **elastic-isotropic**, **plastic-bilinear**, **plastic-multilinear**, **thermo-isotropic**, **thermo-plastic**, **creep**, **plastic-creep**, **multilinear-plastic-creep**, **creep-variable**, **plastic-creep-variable**, **multilinear-plastic-creep-variable**.

- The pipe elements can be used with a **small displacement** or a **large displacement** formulation. In the small displacement formulation, the displacements and rotations are assumed to be infinitesimally small. In the large displacement formulation, the displacements and rotations are assumed to be large. In all cases, only small strains are assumed.

All of the material models listed above can be used with either formulation. The use of a linear material with the small displacement formulation corresponds to a linear formulation, and the use of a nonlinear material with the small displacement formulation corresponds to a materially-nonlinear-only formulation.

In the large displacement formulation, large displacement effects are only included for the overall beam displacements and not for the warping/ovalization degrees of freedom. Hence, large displacement effects are not accounted for in the element cross-sectional deformations.

- An element is to be considered a nonlinear pipe element if a large displacement analysis is performed or a nonlinear material is used and/or pipe internal pressure is present.

2.8.2 Pipe internal pressures

- Both types of pipe elements support internal pressure loading. The load vector corresponding to that pressure is calculated as follows:

First, two axial forces $\mathbf{F}_{a1}^{(p)}$ and $\mathbf{F}_{a2}^{(p)}$ are acting at the element end nodes, as shown in Fig. 2.8-5. Their magnitudes are

$$F_{a1}^{(p)} = \pi a_i^2 p_1 \quad ; \quad F_{a2}^{(p)} = \pi a_i^2 p_2$$

where p_1 and p_2 are the internal pressures at nodes 1 and 2 (input as nodal point pressures to the program), and a_i is the internal radius of the cross-section. Also, if the element is curved, transverse forces are generated by the internal pressure. The transversal force acting on the elementary axial length dL of the pipe axis is

$$d\mathbf{F}_T = -\frac{\pi a_i^2}{R} p s dL$$

and the nodal transversal forces are obtained as

$$\mathbf{F}_k = \int_L h_k d\mathbf{F}_T$$

where h_k is the interpolation function corresponding to node k .

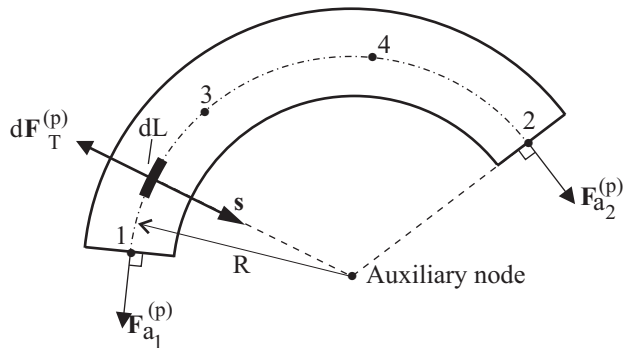
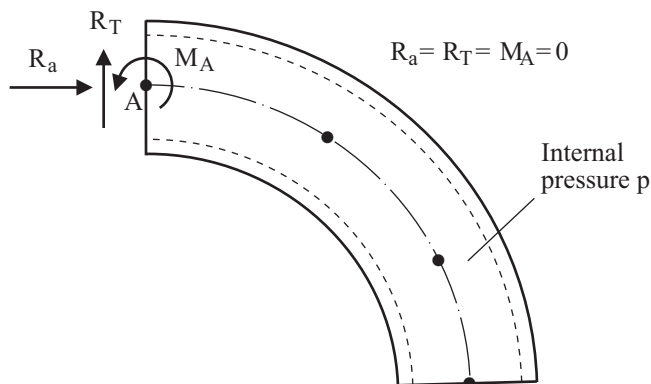
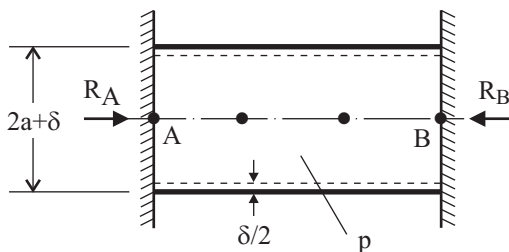


Figure 2.8-5: Forces due to pipe internal pressure

- The nodal forces due to the internal pressure for an element represent a self-balanced system. Consequently, if a structure is free to deform, as in the case of the cantilever in Fig. 2.8-6(a), the support reactions are equal to zero. However, in the case of a restrained system, the reactions at a support balance the forces and moments which are a result of the internal stresses and the nodal forces that are equivalent to the internal pressure, see Fig. 2.8-6(b).
- Note that internal pressure acts against ovalization and hence increases the stiffness of the ovalization degrees of freedom.



a) Cantilever pipe free to deform (zero reactions)



In elastic deformations:

$$R_A = R_B = \pi a_i p (a_i - 2va)$$

$$a_i = a - \delta/2$$

= internal radius

a = mean radius
 v = Poisson's ratio

b) Fully restrained pipe

Figure 2.8-6: Support reactions for pipe loaded by internal pressure

- In the stress calculation the following assumptions are made regarding the internal pressure effects.

► For the pipe-beam element it is assumed that the hoop stress ${}^{t+\Delta t}\sigma_{cc}$ is determined by the given pressure, ${}^{t+\Delta t}\sigma_{cc} = {}^{t+\Delta t}\sigma_{cc}^{(p)}$, where ${}^{t+\Delta t}\sigma_{cc}^{(p)}$ is the hoop stress equivalent to the pressure ${}^{t+\Delta t}p$ (using thin-walled cylinder theory). Then, using that the normal stress through the pipe skin is equal to zero, the axial stress ${}^{t+\Delta t}\sigma_{aa}$ is obtained as

$${}^{t+\Delta t}\sigma_{aa} = {}^{t+\Delta t}E \left({}^{t+\Delta t}e_{aa} - {}^{t+\Delta t}e_{aa}^{IN} - {}^{t+\Delta t}e^{TH} \right) + {}^{t+\Delta t}\nu {}^{t+\Delta t}\sigma_{cc}^{(p)}$$

where ${}^{t+\Delta t}E$ and ${}^{t+\Delta t}\nu$ are the Young's modulus and Poisson's ratio, and ${}^{t+\Delta t}e^{TH}$ is the thermal strain corresponding to the temperature ${}^{t+\Delta t}\theta$; ${}^{t+\Delta t}e_{aa}$ and ${}^{t+\Delta t}e_{aa}^{IN}$ are the total mechanical and inelastic strains, respectively.

- For the pipe-shell element the basic condition regarding the deformation of the pipe skin is that the material is free to deform in the circumferential direction. Hence, the total hoop strain is

$$\begin{aligned} {}^{t+\Delta t}e_{cc} = & -{}^{t+\Delta t}\nu \left({}^{t+\Delta t}e_{aa} - {}^{t+\Delta t}e_{aa}^{IN} \right) - \frac{1}{2} {}^{t+\Delta t}e_{aa}^{IN} + {}^{t+\Delta t}e_{cc}^{OV} \\ & + (1 + \nu) {}^{t+\Delta t}e^{TH} + \left(\frac{1 - \nu^2}{E} \right) \sigma_{cc}^{(p)} \end{aligned}$$

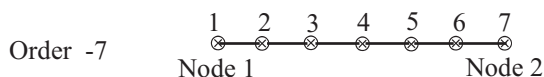
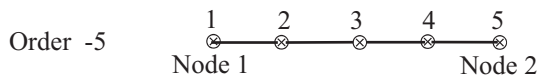
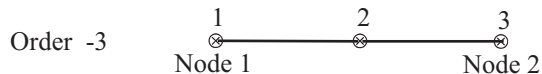
where ${}^{t+\Delta t}e_{cc}^{OV}$ is the strain due to ovalization of the cross-section. This relation is based on the assumption that the inelastic deformation is incompressible (as it is in metal plasticity and creep).

- Pressures at all nodal points of a piping model using the pipe-beam or pipe-shell elements (and other elements) must be defined if internal pressure effects are to be included in some of the elements. The internal pressure loads are only used for the pipe-beam and pipe-shell elements. See Section 5.7 for information about specifying internal pressures.

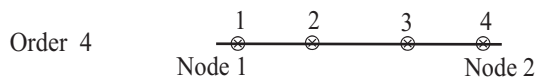
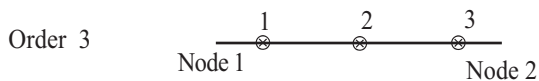
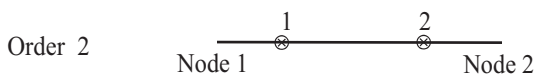
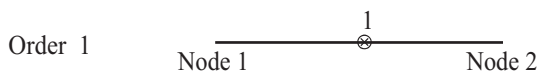
2.8.3 Numerical integration

- The element stiffness and mass matrices and force vectors are in all formulations evaluated using Gauss or Newton-Cotes numerical integration in the axial and the thickness directions and the composite trapezoidal rule along the circumference of the pipe section. The locations and labeling of the integration points are given in Fig. 2.8-7.

Newton-Cotes formulas:

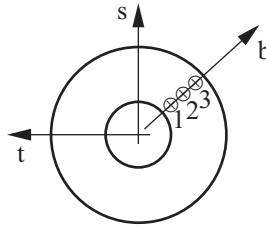


Gauss formulas:

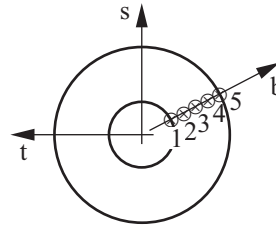


a) Integration point locations on pipe centroidal axis (label INR)

Figure 2.8-7: Locations and labeling of integration points

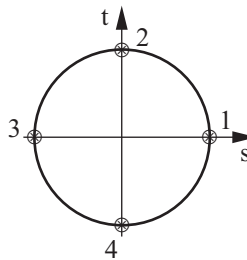


Gauss: order 3

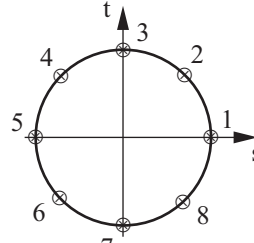


Newton-Cotes: order 5

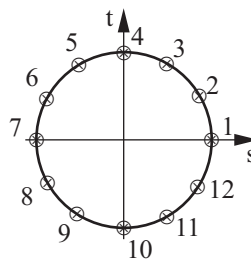
b) Integration point locations along thickness direction (label INB)



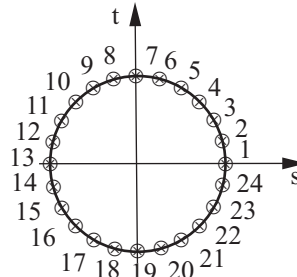
Order 4



Order 8



Order 12



Order 24

c) Integration point locations along circumferential direction (label INC)

Figure 2.8-7: (continued)

- For the pipe-beam elements the following integration orders should be used (and are default in ADINA):
 - ▶ 1-point and 3-point Gauss integration along the axial direction for the 2-node and 4-node elements, respectively
 - ▶ 2-point Gauss integration through the thickness of the pipe skin

- ▶ 8-point integration along the circumference
- For the pipe-shell element the following integration orders should be used (and are default in ADINA):
 - ▶ 3-point Gauss integration along the axial direction
 - ▶ 2-point Gauss integration through the thickness of the pipe skin
 - ▶ 12-point integration along the circumference if only the in-plane ovalization degrees of freedom are activated
 - ▶ 24-point integration along the circumference if the out-of-plane ovalization degrees of freedom are activated
- The recommended numerical integration orders for nonlinear analysis are:
 - ▶ When no inelastic strain effects are considered, use the same integration orders as in linear analysis.
 - ▶ When material nonlinear conditions are considered, use a higher integration order through the pipe skin if the pipe skin progressive yielding or creep is to be represented. For the integration along the length of the elements and the circumference of the cross-section, an integration order higher than that used in linear analysis may, for the same reason, also be appropriate.
 - ▶ The integration order along the axial r-direction (INR) should not be changed from the default value for the pipe-beam element otherwise the pipe-beam element will lock. The same locking phenomenon is also observed for iso-beam elements; see Section 2.5.2 for more details.

2.8.4 Element mass matrices

- The pipe element can be used with a lumped or a consistent mass matrix, except for explicit dynamic analysis which always uses a lumped mass.
- The consistent mass matrix associates mass only with the pipe's translational and rotational degrees of freedom and is calculated using the isoparametric formulation with the displacement interpolations given on p. 408 of ref. KJB. Hence, no masses are

ref. KJB
Section
5.4.1

associated with the ovalization/warping degrees of freedom.

- The lumped mass for translational degree of freedom i is $M \cdot \left(\frac{\ell_i}{L} \right)$ where M is the total element mass, L is the element length and ℓ_i is a fraction of the length associated with node i . The rotational lumped mass is $\frac{1}{3} \cdot M \cdot \left(\frac{\ell_i}{L} \right) \cdot \left(\frac{1}{4} (d_o^2 + d_i^2) \right)$ where d_o and d_i are the outside and inside diameters of the pipe. The lumped ovalization and warping mass for each of the ovalization and warping degrees of freedom are set to be equal to the translational lumped masses $M \cdot \left(\frac{\ell_i}{L} \right)$.
- The rotational lumped mass can be multiplied by a user-specified scalar ETA (except in explicit analysis).

2.8.5 Element output

You can request that ADINA either print or save stresses or forces.

Stresses: Each element outputs, at its integration points, the following information to the porthole file, based on the material model. This information is accessible in the AUI using the given variable names.

Elastic-isotropic: STRESS (ABC) , STRAIN (ABC) ,
FE_EFFECTIVE_STRESS,
EQUIV_INTERNAL_AXIAL_PRESSURE,
EQUIV_INTERNAL_HOOP_PRESSURE

Elastic-isotropic with thermal effects: STRESS (ABC) ,
STRAIN (ABC) , FE_EFFECTIVE_STRESS,
EQUIV_INTERNAL_AXIAL_PRESSURE,
EQUIV_INTERNAL_HOOP_PRESSURE, THERMAL_STRAIN,
ELEMENT_TEMPERATURE

Thermo-isotropic: STRESS (ABC) , STRAIN (ABC) ,
THERMAL_STRAIN (ABC) ,

EQUIV_INTERNAL_AXIAL_PRESSURE,
EQUIV_INTERNAL_HOOP_PRESSURE,
ELEMENT_TEMPERATURE

Plastic-bilinear, plastic-multilinear: PLASTIC_FLAG,
STRESS(ABC), STRAIN(ABC),
PLASTIC_STRAIN(ABC), THERMAL_STRAIN(ABC),
EQUIV_INTERNAL_AXIAL_PRESSURE,
EQUIV_INTERNAL_HOOP_PRESSURE,
FE_EFFECTIVE_STRESS, YIELD_STRESS,
ACCUM_EFF_PLASTIC_STRAIN, ELEMENT_TEMPERATURE

Thermo-plastic, creep, plastic-creep, multilinear-plastic-creep,
creep-variable, plastic-creep-variable, multilinear-plastic-creep-
variable: PLASTIC_FLAG, NUMBER_OF_SUBINCREMENTS,
STRESS(ABC), STRAIN(ABC),
PLASTIC_STRAIN(ABC), CREEP_STRAIN(ABC),
THERMAL_STRAIN(ABC), ELEMENT_TEMPERATURE,
ACCUM_EFF_PLASTIC_STRAIN,
FE_EFFECTIVE_STRESS, YIELD_STRESS,
EQUIV_INTERNAL_AXIAL_PRESSURE,
EQUIV_INTERNAL_HOOP_PRESSURE,
EFFECTIVE_CREEP_STRAIN

In the above lists,

STRESS(ABC) = STRESS-AA, STRESS-AB, STRESS-
AC, STRESS-CC, STRESS-BB

with similar definitions for the other abbreviations used above. See Fig. 2.8-1 for the pipe stress coordinate system convention.

See Section 13.1.1 for the definitions of those variables that are not self-explanatory.

Nodal forces: In linear analysis, the forces/moments at the element nodes are

$$\mathbf{F} = \mathbf{K} \mathbf{U}$$

where \mathbf{K} is the element stiffness matrix and \mathbf{U} is the vector of nodal point displacements which may include ovalizations and warpings. Hence, the forces/moments in \mathbf{F} will be equal to the forces/moments applied externally or by adjoining elements, and

are not necessarily the internal section forces/moments.

The nodal forces are accessible in the AUI using the variable names `NODAL_FORCE-R`, `NODAL_FORCE-S`, `NODAL_FORCE-T`, `NODAL_MOMENT-R`, `NODAL_MOMENT-S`, `NODAL_MOMENT-T`.

The end forces/moments are computed at the element local nodes. In the AUI, element local nodes are defined as element points of type label. For example, to access the result computed at element 5, local node 2, define an element point of type label with element number 5, label number 2.

2.8.6 Recommendations on use of elements

- The pipe elements are available in ADINA to model general pipe networks subjected to general boundary and loading conditions including internal pressure and thermal loading.
- When ovalization effects are insignificant (straight pipes or pipes of large thickness) the 2- or 4-node pipe-beam element should be used. In this case, linear or nonlinear behavior can be modeled with large displacements but small strains.
- When ovalization effects are important, the 4-node pipe-shell element should be used. In this case, linear or nonlinear behavior can be modeled with the assumptions that the displacements are always relatively small and that the strains are small. The displacements are always relatively small because in the calculation of the flexibility due to ovalization/warping the pipe radius is constant (and equal to the initial pipe radius) and no large displacement strain terms are included in the ovalization/warping calculations. The only large displacement strain terms accounted for are those corresponding to the beam behavior.

However, if the large displacement option is employed, the change of the pipe radius is taken into account in the calculation of the element nodal forces due to the pipe internal pressure.

ref. K.J. Bathe and C.A. Almeida, "A Simple and Effective Pipe Elbow Element – Linear Analysis," *J. Appl. Mech., Transactions of the ASME*, Vol. 47, No. 1, pp. 93-100, 1980.

- Since the pipe-beam element stiffness matrix is calculated using numerical integration, it is clear that the linear 2-node Hermitian beam element (see Section 2.4.1) for which the stiffness matrix is evaluated in closed-form can be much more cost-effective in analyzing linear piping systems with negligible ovalization, warping and internal pressure effects.

2.9 General and spring/damper/mass elements

2.9.1 General elements

- General elements are linear elements which can have from one node to as many nodes as there are in the structure.
- General elements are useful in the definition of special elements whose formulations are not directly available in ADINA. General elements are also useful in providing flexibility in obtaining any desirable linear stiffness/mass/damping matrices.
- You directly enter the stiffness matrix (and in dynamic analysis, the mass and damping matrices) of the general element.

The dimension of the stiffness/mass/damping matrices cannot exceed ND, defined as $\text{NDOF} \times \text{IELD}$, where IELD is the number of nodes of the general element and NDOF is the number of active degrees of freedom for each node (set in the MASTER command, with a default of 6). The components for the stiffness/mass/damping matrices must be defined for all ND degrees of freedom. Note that only the upper triangular half of the matrices needs to be input (the stiffness/mass/damping matrices must be symmetric).

For example, for a 4-node 3D solid element with three active master degrees of freedom, $\text{ND} = 3 \times 4 = 12$, the stiffness matrix is 12x12 symmetric and the internal force vector is calculated using

$$\begin{bmatrix} F_x^1 \\ F_y^1 \\ F_z^1 \\ F_x^2 \\ F_y^2 \\ F_z^2 \\ F_x^3 \\ F_y^3 \\ F_z^3 \\ F_x^4 \\ F_y^4 \\ F_z^4 \end{bmatrix} = \begin{bmatrix} k_{1,1} & k_{1,2} & k_{1,3} & k_{1,4} & k_{1,5} & k_{1,6} & k_{1,7} & k_{1,8} & k_{1,9} & k_{1,10} & k_{1,11} & k_{1,12} \\ & k_{2,2} & k_{2,3} & k_{2,4} & k_{2,5} & k_{2,6} & k_{2,7} & k_{2,8} & k_{2,9} & k_{2,10} & k_{2,11} & k_{2,12} \\ & & k_{3,3} & k_{3,4} & k_{3,5} & k_{3,6} & k_{3,7} & k_{3,8} & k_{3,9} & k_{3,10} & k_{3,11} & k_{3,12} \\ & & & k_{4,4} & k_{4,5} & k_{4,6} & k_{4,7} & k_{4,8} & k_{4,9} & k_{4,10} & k_{4,11} & k_{4,12} \\ & & & & k_{5,5} & k_{5,6} & k_{5,7} & k_{5,8} & k_{5,9} & k_{5,10} & k_{5,11} & k_{5,12} \\ & & & & & k_{6,6} & k_{6,7} & k_{6,8} & k_{6,9} & k_{6,10} & k_{6,11} & k_{6,12} \\ & & & & & & k_{7,7} & k_{7,8} & k_{7,9} & k_{7,10} & k_{7,11} & k_{7,12} \\ & & & & & & & k_{8,8} & k_{8,9} & k_{8,10} & k_{8,11} & k_{8,12} \\ & & & & & & & & k_{9,9} & k_{9,10} & k_{9,11} & k_{9,12} \\ & & & & & & & & & k_{10,10} & k_{10,11} & k_{10,12} \\ & & & & & & & & & & k_{11,11} & k_{11,12} \\ & & & & & & & & & & & k_{12,12} \end{bmatrix} \begin{bmatrix} u_x^1 \\ u_y^1 \\ u_z^1 \\ u_x^2 \\ u_y^2 \\ u_z^2 \\ u_x^3 \\ u_y^3 \\ u_z^3 \\ u_x^4 \\ u_y^4 \\ u_z^4 \end{bmatrix}$$

sym

The stiffness matrix is entered using the following command:

```

MATRIX STIFFNESS 1 12
1 k1,1 k1,2 k1,3 k1,4 k1,5 k1,6 k1,7 k1,8 k1,9 k1,10 k1,11 k1,12
2 k2,2 k2,3 k2,4 k2,5 k2,6 k2,7 k2,8 k2,9 k2,10 k2,11 k2,12
3 k3,3 k3,4 k3,5 k3,6 k3,7 k3,8 k3,9 k3,10 k3,11 k3,12
4 k4,4 k4,5 k4,6 k4,7 k4,8 k4,9 k4,10 k4,11 k4,12
5 k5,5 k5,6 k5,7 k5,8 k5,9 k5,10 k5,11 k5,12
6 k6,6 k6,7 k6,8 k6,9 k6,10 k6,11 k6,12
7 k7,7 k7,8 k7,9 k7,10 k7,11 k7,12
8 k8,8 k8,9 k8,10 k8,11 k8,12
9 k9,9 k9,10 k9,11 k9,12
10 k10,10 k10,11 k10,12
11 k11,11 k11,12
12 k12,12

```

where, of course, the numerical values of the stiffness matrix entries are substituted for k_{ij} . Note that in this case, all except for the three translational degrees of freedom must be disabled from the MASTER command.

- An alternative is to define a user-supplied general element, see Section 2.9.3.

If skew systems are used, then either all matrices must be input in the global coordinate system and proper transformations are made by ADINA if there are skew systems at the element group nodes, or

all matrices must be input in the coordinate systems used at the nodes and no transformations between global and skew systems are made in ADINA (SKEWSYSTEMS = YES in the EGROUP GENERAL or EGROUP SPRING command).

- Note that the general element can be used to define a constant damping matrix for other element types in ADINA. Alternative ways to define damping are Rayleigh damping, concentrated dampers and the damping feature in spring elements.
- Forces are calculated for general elements using

$$\mathbf{F} = \mathbf{K} \mathbf{U}$$

where \mathbf{U} stores the displacements and rotation of the ND degrees of freedom.

The forces are accessible in the AUI using the variable names NODAL_FORCE-X, NODAL_FORCE-Y, NODAL_FORCE-Z, NODAL_MOMENT-X, NODAL_MOMENT-Y, NODAL_MOMENT-Z. The forces are with reference to the nodal skew systems, if skew systems are used.

- The damping force is simply calculated as

$$\mathbf{F} = \mathbf{C} \dot{\mathbf{U}}$$

and the inertia is calculated as

$$\mathbf{F} = \mathbf{M} \ddot{\mathbf{U}}$$

- For output purposes, stresses can also be calculated for general elements if you define a stress-displacement matrix \mathbf{S} . The stress vector $\boldsymbol{\sigma}$ is obtained using

$$\boldsymbol{\sigma} = \mathbf{S} \mathbf{U}$$

Note that the matrix \mathbf{S} must be input as a full matrix and is of dimension NS×ND, where NS equals the number of stress components in $\boldsymbol{\sigma}$ (NS cannot exceed 600).

Each element outputs the vector σ to the porthole file. This vector is accessible in the AUI using the variable name `GENERAL_ELEMENT_STRESS`. You select the component of σ in the AUI by setting the label point number equal to the component number in σ .

2.9.2 Linear and nonlinear spring/damper/mass elements

Linear spring single-degree-of-freedom element

- The stiffness matrix has only 1 component and corresponds to a grounded spring acting in the single (user-specified) degree of freedom, see Fig. 2.9-1(a).
- The mass matrix corresponds to a concentrated mass acting in the single degree of freedom, see Fig. 2.9-1(b).
- The damping matrix corresponds to a grounded damper acting in the single degree of freedom, see Fig. 2.9-1(c).

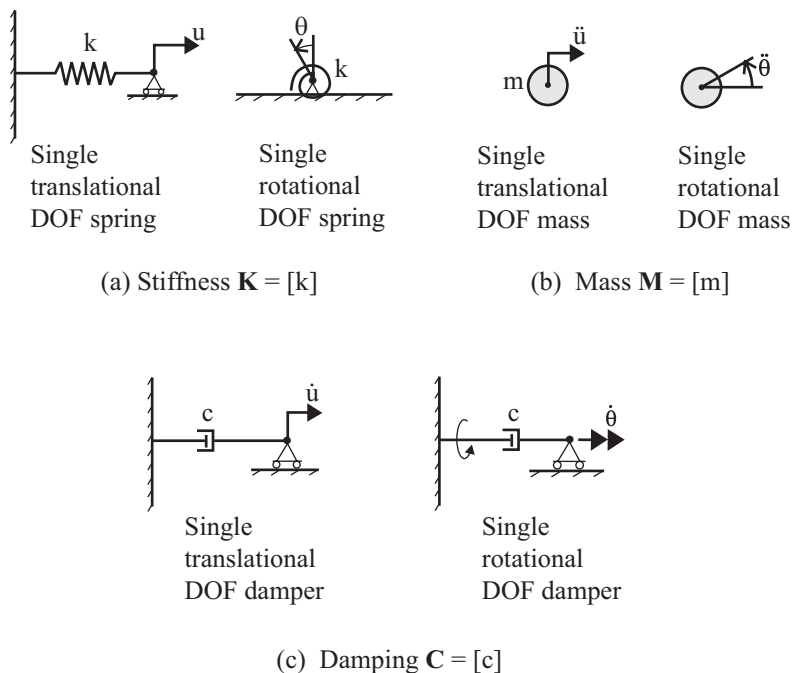


Figure 2.9-1: Single-degree-of-freedom element

Linear spring two-degree-of-freedom element

- The stiffness matrix is shown in Fig. 2.9-2(a) and corresponds to a spring coupling the two (user-specified) degrees of freedom.
- The lumped mass matrix is diagonal, see Fig. 2.9-2(b), and corresponds to two concentrated masses acting in the two degrees of freedom.
- The consistent mass matrix is obtained using a linear interpolation of the accelerations in the two degrees of freedom, see Fig. 2.9-2(b) and ref. KJB, Example 4.5, pp. 166-170.
- The damping matrix is shown in Fig. 2.9-2(c) and corresponds to a damper coupling the two degrees of freedom.

MNO-G spring element

- The element definition is the same as that of the single- or two-

degree of freedom linear spring element.

- Spring stiffness can be defined using a nonlinear force-displacement curve as in the nonlinear spring element.
- Spring stiffness can be in any direction defined by the command
EGROUP SPRING . . . SKEWSYSTEM=YES
- Damping coefficient can be dependent on the relative velocity of the element end nodes as in the nonlinear damper element.
- Element mass can vary with time as in the nonlinear mass element.

$$\mathbf{K} = \begin{bmatrix} k & -k \\ -k & k \end{bmatrix} \quad \mathbf{M}_{\text{lumped}} = \begin{bmatrix} m/2 & 0 \\ 0 & m/2 \end{bmatrix}$$

k = spring constant

a) Stiffness matrix

$$\mathbf{M}_{\text{consistent}} = \begin{bmatrix} m/3 & m/6 \\ m/6 & m/3 \end{bmatrix}$$

m = total mass of element

$$\mathbf{C} = \begin{bmatrix} c & -c \\ -c & c \end{bmatrix}$$

b) Mass matrix

c = damping constant

c) Damping matrix

Figure 2.9-2: Two-degrees-of-freedom element matrices

Nonlinear spring/damper/mass element

- A nonlinear "spring" element is available in ADINA for static and dynamic analysis. This element can be used to connect two nodes or to attach a node to the ground. It can be a translational or a rotational "spring" (see Figure 2.9-3). The stiffness, damping and mass of the element can each be either nonlinear or linear.

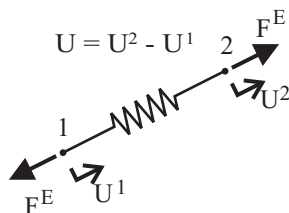
- Large displacements can be applied to nonlinear spring elements.
- In the case of a translational spring, the nonlinear stiffness is defined by a multilinear elastic force versus relative-displacement relationship. The corresponding curve can be different for negative relative-displacement than for positive relative-displacement. The input curve need not pass through the origin (0,0).

The nonlinear damping is given by a function of the form

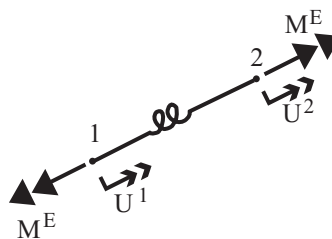
$|F^D| = C|\dot{U}|^N$ where F^D is the damping force in the element, \dot{U} is the relative velocity between the element end nodes, and C and N are real constants.

The mass of the element can vary with time (see Figure 2.9-4).

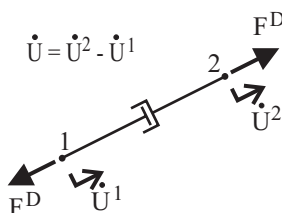
Nonlinear translational stiffness:



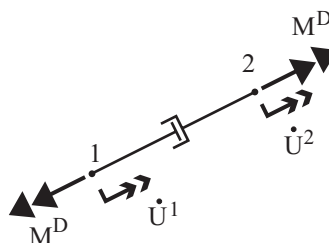
Nonlinear torsional stiffness:



Nonlinear translational damping:



Nonlinear torsional damping:



Nonlinear translational stiffness,
grounded spring:

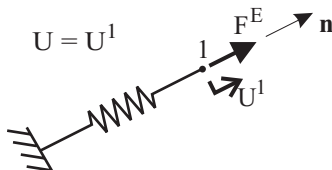
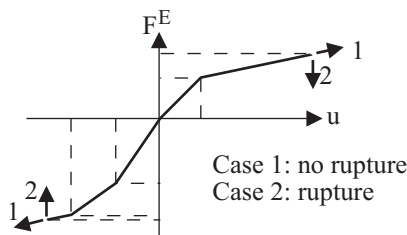


Figure 2.9-3: Description of the nonlinear spring/damper/mass element

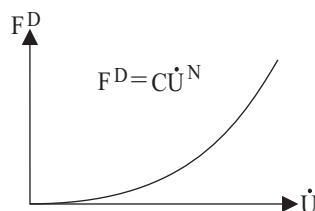
The relative displacement \mathbf{U} is defined as follows: Let \mathbf{U}^1 and \mathbf{U}^2 be the displacement vectors at local nodes 1 and 2, and ${}^0\mathbf{x}^1$ and ${}^0\mathbf{x}^2$ be the respective initial positions of these nodes.

For initially coincident nodes (${}^0\mathbf{x}^1 = {}^0\mathbf{x}^2$), $U = U_{id}^2 - U_{id}^1$ where id specifies a global direction in the global Cartesian coordinate system, and is specified in the input. $id = 1, 2, 3$ for the X, Y, Z translational directions and $id = 4, 5, 6$ for the rotation about the X, Y, Z directions. There is also an option (using $id = 0$) to use another arbitrary direction.

Force versus displacement or
moment versus rotation:



Force versus velocity or moment
versus torsional velocity:



Mass versus time:

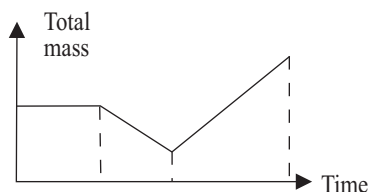


Figure 2.9-4: Nonlinear stiffness, damping and mass

If the nodes are not initially coincident, then the element can be used to define a translational spring along the direction given by the vector from node 1 to node 2. Let \mathbf{U}^I store only the translational displacements, then the relative translational

displacement is given by
$$U = (\mathbf{U}^2 - \mathbf{U}^1) \cdot \frac{\begin{pmatrix} {}^0\mathbf{x}^2 - {}^0\mathbf{x}^1 \end{pmatrix}}{\left\| \begin{pmatrix} {}^0\mathbf{x}^2 - {}^0\mathbf{x}^1 \end{pmatrix} \right\|}, \text{ i.e., the}$$

relative displacement is the difference in displacements in the direction of the line (spring) element.

- In a similar manner, a torsional spring is defined between the local nodes 1 and 2.
- Similar definitions are used for the relative velocity.
- Rupture based on the value of the relative displacement may or may not be included in the nonlinear spring element (see Figure 2.9-4). If rupture is not desired, the last segments of the elastic-force versus relative displacement curve are extrapolated when necessary. If rupture is desired, then the elastic force in the element is set to zero whenever the relative displacement is larger (resp. smaller) than the last (resp. first) point on the curve. Note

that this effect is reversible, as the force will be computed again according to the curve if the displacement value comes back into the curve displacement range.

- Note that the linear spring element and the nonlinear spring element act quite differently: the linear spring element connects specific degrees of freedom, whereas the nonlinear spring element connects nodes and acts in a specified direction. Likewise the input for these elements is also different.

Element output

- You can request that ADINA either print or save stress resultants or forces.

Stress resultants: Each element outputs the following information to the porthole file. This information is accessible in the AUI using the given variable names.

Linear element: GENERAL_ELEMENT_STRESS (NS values per element)

Nonlinear element: DAMPING_FORCE, ELASTIC_FORCE

Nodal forces: For linear elements, these are the elastic nodal forces (in the global Cartesian reference system) equivalent to the axial elastic stress. Any stresses due to damping are not included. For nonlinear elements, these are the nodal forces (in the global Cartesian reference system) equivalent to the axial elastic and damping stresses.

The forces are accessible in the AUI using the variable names NODAL_FORCE-X, NODAL_FORCE-Y, NODAL_FORCE-Z, NODAL_MOMENT-X, NODAL_MOMENT-Y, NODAL_MOMENT-Z. The forces are with reference to the nodal skew systems, if skew systems are used.

2.9.3 6DOF Spring Element

- The 6DOF spring element is a generalized spring-damper element which can be linear or materially-nonlinear only (MNO). It can have single node, two coincident or two non-coincident nodes. In each degree of freedom, the element stiffness can be defined as a

constant or using a force-displacement curve in the element coordinate system. The damping coefficients are always constants in units of force per unit velocity. Please note that mass property is not available in the 6DOF spring element. In the case of the MNO 6DOF spring, the stiffness is defined by a multilinear elastic force versus relative-displacement curve. The input curve can be different for negative relative-displacement than for positive relative-displacement. The input curve need not pass through the origin (0.0).

- The displacement (skew) system can be defined in the 6DOF spring element to prescribe loads and constraints. Element birth/death is also supported. Currently, the 6DOF spring element is not supported in explicit dynamic analysis.
- If a 6DOF spring element has a single node or two coincident nodes, its element coordinate system must be defined using CID (coordinate system used to define spring element) as shown in Fig. 2.9-5 and Fig. 2.9-6. For single node 6DOF spring element, it corresponds to a grounded spring acting in the user-specified degree of freedom.

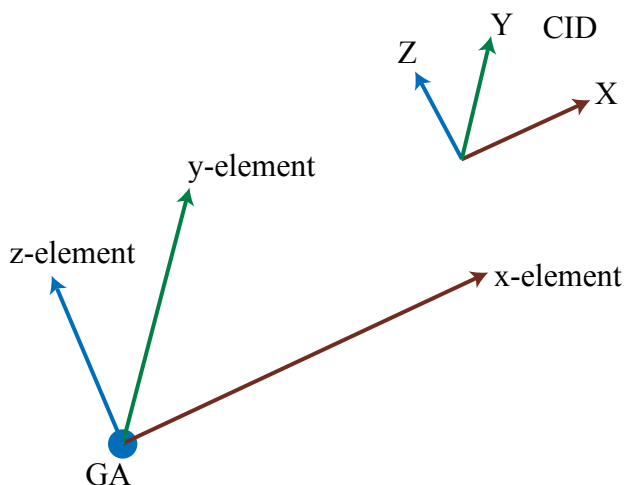


Figure 2.9-5: 6DOF spring element with single node

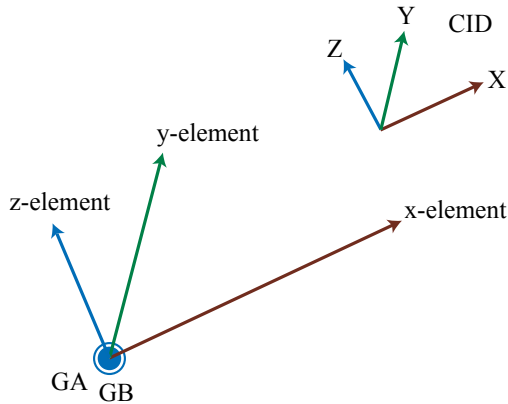


Figure 2.9-6: 6DOF spring element with two coincident nodes

If a 6DOF spring element has two non-coincident nodes, its element coordinate system can be defined using CID, orientation vector or its axial direction as shown in Fig. 2.9-7, Fig. 2.9-8 and Fig. 2.9-9.

In Fig. 2.9-7, the element coordinate system is defined by CID. Note that GA and GB may or may not have displacement (skew) coordinate system.

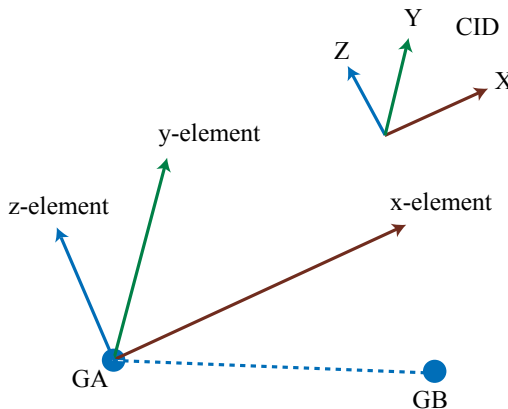


Figure 2.9-7: 6DOF spring element with two non-coincident nodes

In Fig. 2.9-8, the element coordinate system is defined by orientation vector using $G0$ or (Ax, Ay, Az) . Note that (Ax, Ay, Az) refers to the displacement (skew) coordinate system of GA .

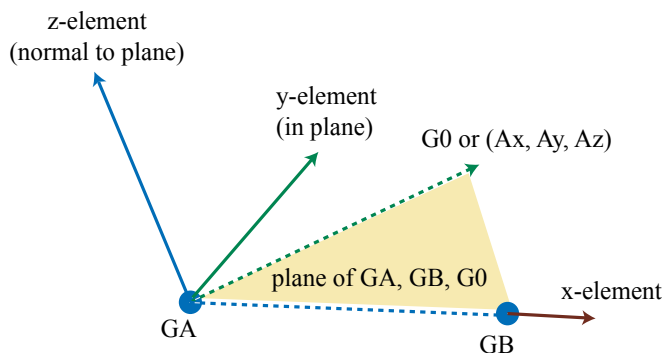


Figure 2.9-8: 6DOF spring element with two non-coincident nodes

In Fig. 2.9-9, a 6DOF spring element is defined with two non-coincident nodes without $G0$, (Ax, Ay, Az) or CID . This defines a 1-D axial/torsional spring/damper. In this case, axial stiffness (or damping) or torsional stiffness (or damping) or both must be specified but all other stiffness (or damping) must not be specified.



Figure 2.9-9: 1-D spring element with two non-coincident nodes

Element output

You can request that ADINA either print or save stress resultants or forces.

Stress resultants: Each element outputs the following information

to the porthole file. This information is accessible in the AUI using the given variable names.

STRESS_TRANS_R, STRESS_TRANS_S, STRESS_TRANS_T,
STRESS_ROT_R, STRESS_ROT_S, STRESS_ROT_T

The above results are always given in the element local coordinate system (r,s,t).

Nodal forces: the forces (in the global Cartesian coordinate system) are accessible in the AUI using the given variable names.

NODAL_FORCE-X, NODAL_FORCE-Y, NODAL_FORCE-Z,
NODAL_MOMENT-X, NODAL_MOMENT-Y,
NODAL_MOMENTZ

2.9.4 User-supplied element

- This feature is useful for implementing elements that have a non-standard behavior such as an element that has different axial and bending parameters, a nonlinear spring for which the force is described as a nonlinear function of the displacement, among other cases.
- The non-standard element behavior must be coded in the ADINA subroutine CUSERG. Subroutine CUSERG must calculate the element stiffness matrix and nodal force vector. In certain types of analysis (e.g., in a dynamic analysis), the mass and damping matrices must also be calculated. Subroutine CUSERG can also output the desired results into the corresponding output files.
- A template and example for the user-supplied element subroutine CUSERG is provided in file `ov1160u.f`. The mentioned example will be discussed later in this section and corresponds to a three-dimensional nonlinear translational spring that takes into account geometric nonlinearities. Also, a `Makefile` file is provided, which includes instructions for compiling and linking the user-supplied element subroutine to ADINA.

- The following element types are allowed:
 - ▶ Triangular 3-node, 6-node, or 7-node 2-D solid element
 - ▶ Quadrilateral 4-node, 8-node, or 9-node 2-D solid element
 - ▶ Tetrahedral 4-node, 10-node, or 11-node 3-D solid element
 - ▶ Hexahedral 8-node, 20-node, 21-node, or 27-node 3-D solid element
 - ▶ 2-node, 3-node, or 4-node iso-beam element
 - ▶ 4-node, 8-node, 9-node, or 16-node shell element
 - ▶ transition shell element with variable number of nodes up to a maximum of 32 nodes.

The nodal connectivity of the user-supplied elements must be compatible with the ordinary ADINA elements so that the user-supplied elements can be generated by the AUI mesh generation commands. This will allow the AUI to load and display the user-supplied elements for post-processing purposes.

- There are two ways within the AUI to activate the user-supplied element feature. Note that regardless of the way chosen, the material definition of the element can only be done with the MATERIAL USER-SUPPLIED command (see AUI Command Reference Manual). The material properties can be temperature dependent or independent. However, this does not imply that a user-supplied material subroutine is needed. In fact, any material implementation needed for the user-supplied element must be done within the user-supplied element subroutine CUSERG.
- The first way to activate the user-supplied element feature consists of basically two parts: material definition and element group definition. The element group definition consists of applying either the EGROUP TWOSOLID or EGROUP THREESOLID or EGROUP ISOBEAM or EGROUP SHELL commands with the parameter OPTION=USER-CODED (see AUI Command Reference Manual). Clearly, the choice of element group command depends

on the type of element the user-supplied element resembles. This is the preferred way.

- The second way to activate the user-supplied element feature consists of applying the general element strategy, which has four parts: material definition, matrix definition, matrix set definition and element group definition.

As with any general element, the stiffness, mass and damping matrices must be provided (of course, the mass and damping matrices are necessary only for certain types of analysis). The stiffness, mass and damping matrices are grouped into a matrix set and the matrix set number is included in the element group definition. In ordinary general elements (see Section 2.9.1), the user directly specifies the above matrices. By implementing the user-supplied element as a general element one can provide the above matrices by either calculating all of them in subroutine CUSERG or by calculating at least the stiffness matrix in subroutine CUSERG and directly specifying the rest as it is done for ordinary general elements.

The element subtype, material label and number of integration points are provided to ADINA through a matrix definition. Note that in the AUI the matrix definition is accessed as a user-supplied matrix from the Matrix Set dialog box. The matrix definition has the following parameters:

- ▶ The element subtype to which the user-supplied element resembles. The options are 2-D solid, 3-D solid, iso-beam or shell.
- ▶ The user-supplied material number.
- ▶ The total number of user-supplied integration points NUIPT. The default value of NUIPT is $NUIT1 \cdot NUIT2 \cdot NUIT3$.
- ▶ The integration order in the first local direction NUIT1. If NUIPT is entered, then NUIT1 is ignored.
- ▶ The integration order in the second local direction NUIT2. If NUIPT is entered, then NUIT2 is ignored.

- The integration order in the third local direction NUIT3. If NUIPT is entered, then NUIT3 is ignored.

- It is important to highlight that the number of active degrees of freedom per node in a user-supplied element is defined by the IDOF parameter of the MASTER command (see AUI Command Reference Manual). This fact may prevent users from creating models that combine user-supplied elements with standard ADINA elements which possess different active degrees of freedom per node.

The standard number of active degrees of freedom per node are 2, 3 and 6 for the 2-D solid, 3-D solid and iso-beam elements, respectively. Note that six degrees of freedom per node are always assumed for the shell user-supplied element.

- The nodes per individual user-supplied element can be assigned with the ENODES command. In addition, the COORDINATES NODE and NODESET commands can be useful when implementing a model with user-supplied elements. Please see the AUI Command Reference Manual for a description of these commands.

- The calls from ADINA to subroutine CUSERG are divided into several phases, which are controlled by the integer variable KEY.

KEY = 1: This is the input phase and it is called once at the beginning of the analysis. The working arrays ARRAY(LGTH1,NUIPT) and IARRAY(LGTH2,NUIPT), which must store real and integer history dependent variables (for all the integration points of an element), respectively, must be initialized to their proper values. Also the coordinates of the integration points of an element XYZIPT(3,NUIPT) must be provided. These points are used for storing, e.g., stress and strain results. The XYZIPT(3,NUIPT) array is arranged through the sequence of 1 to NUIPT or through the local integration orders NUIT1, NUIT2 and NUIT3. The definitions of NUIT1, NUIT2 and NUIT3 follow the standard ADINA (2-D solid, 3-D solid, iso-beam and shell) elements local integration orders. Note that the array XYZ(3,IELD) provides the element nodal coordinates and can be used to calculate the requested integration point coordinates.

KEY = 2: The element nodal forces $RE(ND)$ must be calculated in the global coordinate system, where ND is the total number of degrees of freedom in an element. For this purpose, ADINA provides the element displacements $DISP2(ND)$ and $DISP1(ND)$ at the current and previous time steps, respectively. In addition, material properties and other parameters are provided (see file `ov1160u.f`), which can be useful. Note that the element nodal forces are the equilibrating forces during the equilibrium iterations and the resultant nodal forces when a solution converges. If the element subtype is iso-beam, then the element nodal forces in the local coordinate system $REBM(ND)$ must also be provided.

In this phase, it is recommended to calculate the desired user-calculated quantities for all the integration points of an element, such as stresses, strains, stress state flag, etc. Depending on the requested element group results (i.e., forces or stresses) this might be a necessary step since ADINA writes the requested results to the porthole file for post-processing. The real and integer user-calculated quantities are stored in $RUPLOT(100,NUIPT)$ and $IUPLOT(50,NUIPT)$, respectively. In order for ADINA to correctly process the information stored in these arrays, the user should follow the order specified in file `ov1160u.f`.

Finally, the real and integer history dependent variables (for all the integration points of an element) stored in $ARRAY(LGTH1,NUIPT)$ and $IARRAY(LGTH2,NUIPT)$, respectively, must be updated.

KEY = 3: The element stiffness matrix $AS(ND,ND)$ must be calculated in this phase. It should be calculated in the global coordinate system. As in the case of the element nodal forces, the stiffness matrix can also be calculated via the element displacements, material properties, and other variables and parameters provided by the ADINA subroutine `CUSERG`.

KEY = 4: The user printout must be coded in this phase. In order for ADINA to access it, use the `AUI` command `PRINTOUT` in your input file.

KEY = 5: In this phase, the array $AS(ND,ND)$ stores the element mass matrix, which must be calculated in the global coordinate system and in terms of the variables and parameters provided by the ADINA subroutine `CUSERG`.

KEY = 6: In this phase, the array $AS(ND,ND)$ stores the element damping matrix, which must be calculated in the global coordinate system and in terms of the variables and parameters provided by the ADINA subroutine CUSERG.

- A complete description of the argument list of subroutine CUSERG is provided in file `ov1160u.f`.
- The following steps provide a guide for using the user-supplied element feature via the AUI commands:

1. Once the user-supplied element has been coded in file `ov1160u.f`, compile and link it to ADINA with the help of the provided `Makefile`.
2. Use the MATERIAL USER-SUPPLIED command to define the material properties, control parameters and working array sizes.
3. If following the preferred way, then use either the EGROUPTWODSOLID or EGROUPTHREEDSOLID or EGROUPISOBEAM or EGROUSHELL commands with the parameter `OPTION=USER-CODED` to define the type of user-supplied element.

If following the general element strategy, then

- (a) use the MATRIX USER-SUPPLIED command to define the type of user-supplied element to relate to the stiffness, mass and damping matrices. Note that if the mass or damping matrices are constants, then they can be input directly with the MATRIX MASS or MATRIX DAMPING commands, respectively;
- (b) use the MATRIXSET command to define the general element matrix set; and
- (c) use the EGROUPTGENERAL command with the label number corresponding to the defined MATRIXSET

and the parameter USER-SUPPLIED=YES to conclude the user-supplied element definition.

4. Define the number of active degrees of freedom per node in the user-supplied element with the IDOF parameter of the MASTER command.
5. Define the rest of the input file (i.e., nodes, nodal connectivity, boundary conditions, loads, etc.) with the corresponding AUI commands and generate the data file. Run ADINA to solve the problem and use the AUI to display the results.

2.9.4.1 User-supplied element example

- A three-dimensional nonlinear translational spring that takes into account geometric nonlinearities is given as an example of the user-supplied element subroutine CUSERG, which can be found in file `ov1160u.f`.
- The spring can be implemented with either the EGROUP ISOBEAM or EGROUP GENERAL AUI commands, and will have three translational degrees of freedom active per node.
- Assume that initially the spring is undeformed and that nodes 1 and 2 are attached to the ends of the spring, then its undeformed length is given by ${}^0L = \left\| {}^0\mathbf{x}^2 - {}^0\mathbf{x}^1 \right\|$.
- The spring force is a function of the displacement, that is, $\mathbf{F} = \mathbf{F}(\mathbf{u})$. Consequently, at the next time step (say at time t) the force at node 1 can be written as

$${}^t\mathbf{F}^1 = k(\Delta L) \frac{\Delta L}{{}^tL} \left({}^t\mathbf{x}^1 - {}^t\mathbf{x}^2 \right) \quad (2.9-1)$$

where ${}^t\mathbf{x}^i = {}^0\mathbf{x}^i + \mathbf{u}^i$ and \mathbf{u}^i is the displacement vector of the i -th node, $\Delta L = {}^tL - {}^0L$, and, in this particular example, it will be assumed that

$$k(\Delta L) = k_0 + k_1 \Delta L \quad (2.9-2)$$

with k_0 and k_1 being spring constants.

- Note that the force at node 2 is ${}^t\mathbf{F}^2 = -{}^t\mathbf{F}^1$.
- The stiffness matrix of the user-supplied element can be easily obtained from the above equations by taking the partial derivative of each nodal force vector with respect to all the degrees of freedom of the user-supplied element. In this particular example, the stiffness matrix is a 6 x 6 symmetric matrix.

2.10 Displacement-based fluid elements

- The elements discussed in this section incorporate the following assumptions:
 - ▶ Inviscid, irrotational medium with no heat transfer
 - ▶ Compressible or almost incompressible medium
 - ▶ Relatively small displacements
 - ▶ No actual fluid flow
- For other types of fluid flow, use ADINA-F and ADINA-FSI (see the ADINA-F Theory and Modeling Guide for details of the formulations employed). In the remainder of this section, only the ADINA fluid elements are discussed.
- The types of problems for which the ADINA displacement-based fluid elements can be employed are:
 - ▶ Static analyses, where the pressure distribution in the fluid and the displacement and stress distribution in the structure is of interest
 - ▶ Frequency analyses, where natural frequencies and mode shapes of a structure/fluid medium are to be determined

- ▶ Transient analyses, where a pressure wave propagates rapidly through the fluid, which does not undergo large motions (transient acoustic problems)

However, in practice, the use of the displacement-based elements is rather restricted to special applications in static and dynamic analyses. The potential based element (see Section 2.11) is much more general and is recommended for use for solving this class of problems.

- Simply stated, the displacement-based fluid elements can be thought of as derived from the solid two- and three-dimensional elements (see Sections 2.2 and 2.3) by using an elastic stress-strain relation with a bulk modulus K and a zero shear modulus.
- The elements can be employed in 2-D and 3-D analyses. Two-dimensional fluid elements can be employed in planar and axisymmetric analyses. Two-dimensional fluid elements must be defined in the YZ plane, and axisymmetric elements must lie in the +Y half plane (all nodal point coordinates must have positive y values).
- Although the fluid elements can be employed using a large displacement formulation, the allowed fluid motion is relatively small in a practical analysis, because the elements must not become distorted. Actual flow of a fluid cannot be analyzed using the elements. Use ADINA-F or ADINA-FSI if the allowed fluid motion is large.
- Difficulties in the use of the elements and various experiences in solutions obtained are discussed in the following references:

- ref. K.J. Bathe and W. Hahn, "On Transient Analysis of Fluid-Structure Systems," *Computers & Structures*, Vol. 10, pp. 383-391, 1981.
- ref. J. Sundqvist, "An Application of ADINA to the Solution of Fluid-Structure Interaction Problems," *Computers & Structures*, Vol. 17, pp. 793-808, 1983.
- ref. L. Olson and K.J. Bathe, "A Study of Displacement-

Based Fluid Finite Elements for Calculating Frequencies of Fluid and Fluid-Structure Systems," *Nuclear Engineering and Design*, Vol. 76, pp. 137-151, 1983.

For example, in Olson and Bathe, it is noted that problems involving structures moving through fluids that behave almost incompressibly (e.g., an ellipse vibrating on a spring in water) cannot be solved satisfactorily with the displacement-based fluid elements.

- In linear analysis, you can impose irrotational conditions in the element formulation. This is achieved using a penalty constraint. Note that if the penalty constraint is imposed, rigid body rotations of the elements are no longer possible.
- These elements are defined within 2-D and 3-D fluid element groups (in ADINA). Set the formulation of the element group to either displacement-based without rotation penalty or displacement-based with rotation penalty (if you are using the AUI user interfaces, set the formulation with the "interpolation type" field).
- Pressures are output at the integration points. The integration point numbering is the same as the numbering convention used for the solid elements, see Sections 2.2.3 and 2.3.3.

The pressure is evaluated using the following relation:

$$p = -K e_v$$

where p is the pressure, K is the bulk modulus and e_v is the volumetric strain (Δ volume/volume).

The pressures and strain components are accessible in the AUI using the variable names FE_PRESSURE, STRAIN-XX, STRAIN-YY, STRAIN-ZZ, STRAIN-XY, STRAIN-XZ, STRAIN-YZ (STRAIN-XY and STRAIN-XZ are applicable only for 3-D elements). See Section 13.1.1 for the definitions of those variables that are not self-explanatory.

- You can also request nodal force output. The nodal forces are accessible in the AUI using the variable names NODAL_FORCE-X, NODAL_FORCE-Y, NODAL_FORCE-Z (NODAL_FORCE-X is

applicable only for 3-D elements).

2.11 Potential-based fluid elements

- The elements discussed in this section incorporate the following assumptions:
 - ▶ Inviscid, irrotational medium with no heat transfer
 - ▶ Compressible or almost incompressible medium
 - ▶ Relatively small displacements of the fluid boundary
 - ▶ Actual fluid flow with velocities below the speed of sound (subsonic formulation) or no actual fluid flow (linear infinitesimal velocity formulation)
- For other types of fluid flow, use ADINA-F and ADINA-FSI (see the ADINA-F Theory and Modeling Guide for details of the formulations employed). In the remainder of this section, only the ADINA fluid elements are discussed.
- The types of problems for which the ADINA potential-based fluid elements can be employed are:
 - ▶ Static analyses, where the pressure distribution in the fluid and the displacement and stress distribution in the structure is of interest
 - ▶ Frequency analyses, where natural frequencies and mode shapes of a structure/fluid medium are to be determined
 - ▶ Transient analyses, where a pressure wave propagates rapidly through the fluid, which does not undergo large motions (transient acoustic problems)
 - ▶ Transient analyses, where fluid flows through the domain, and the boundaries of the domain undergo only small motions.
- The potential-based fluid elements can be employed in 2-D and 3-D analyses. Two-dimensional elements can be employed in

planar and axisymmetric analyses. Two-dimensional elements must be defined in the YZ plane, and axisymmetric elements must lie in the +Y half plane (all nodal point coordinates must have non-negative y values).

- The potential-based fluid elements can be coupled with ADINA structural elements, as described in detail below. The structural motions cause fluid flows normal to the structural boundary, and the fluid pressures cause additional forces to act on the structure.
- The potential-based fluid elements can be coupled to a pressure boundary condition (i.e., no structure adjacent to the potential-based fluid element boundary). This feature can be used to model free surfaces.
- The potential-based fluid elements can be coupled directly to ADINA-F fluid elements. The ADINA-F fluid element motions cause potential-based fluid flows normal to the ADINA-F boundary, and the potential-based fluid pressures cause additional forces to act on the ADINA-F boundary.
- The potential-based elements can model (approximately) an infinite domain through the use of special infinite elements.
- Two formulations are available, a subsonic velocity formulation, which is nonlinear, and an infinitesimal velocity formulation, which is linear. These are described in detail in the following sections.
- In some analyses, either ADINA-F/ADINA-FSI or the potential-based fluid elements can be used in the modeling. ADINA-F/ADINA-FSI is far more general than the potential-based fluid elements and can model a much wider range of flow conditions. In addition, in ADINA-FSI, the fluid mesh need not be compatible with the structural mesh.

However, for the class of problems in which the assumptions given above are acceptable, the potential-based fluid elements are more efficient. This is because

- ▶ The number of degrees of freedom in the fluid region is less for the potential-based formulation. In 3-D analysis, each

ADINA-F node requires a minimum of 4 degrees of freedom, whereas each node in the interior of a potential-based mesh requires only one degree of freedom.

- ▶ If the velocities are small, then the linear infinitesimal velocity formulation can be used. ADINA-F is always nonlinear.
- ▶ In addition, frequency analysis is not possible in ADINA-F.
- The potential-based element formulation has been extensively revised in version 8.0. Some of the features of earlier versions of ADINA are superseded by new features of ADINA 8.0. Features of earlier versions of ADINA that are superseded by new features of ADINA 8.0 include:

The P_0 degree of freedom (superseded in ADINA 7.4)

Volume infinite elements (superseded in ADINA 8.0)

These features are retained in ADINA 8.0 so that models developed for earlier versions of ADINA still work in ADINA 8.0. However these features are not described here, and are not recommended for new models.

- For another description of the potential-based fluid element, including examples, see the following reference:

ref. T. Sussman and J. Sundqvist, "Fluid-structure interaction analysis with a subsonic potential-based fluid formulation," *Computers & Structures*, Vol. 81 (2003), pp. 949-962.

2.11.1 Theory: Subsonic velocity formulation

- Figure 2.11-1 shows a fluid region with volume and bounding surface. In the fluid, we use the basic equations of continuity and energy/momentum, as written in terms of the velocity potential:

$$\dot{\rho} + \nabla \cdot (\rho \nabla \phi) = 0 \quad (2.11-1)$$

and

$$h = \Omega(\mathbf{x}) - \dot{\phi} - \frac{1}{2} \nabla \phi \cdot \nabla \phi \quad (2.11-2)$$

where ρ is the density, ϕ is the velocity potential ($\mathbf{v} = \nabla \phi$ where \mathbf{v} is the fluid velocity), h is the specific enthalpy (defined as $h = \int \frac{dp}{\rho}$), p is the pressure and $\Omega(\mathbf{x})$ is the potential of the (conservative) body force accelerations at position \mathbf{x} . For example, when the body forces are due to gravity, $\nabla \Omega = \mathbf{g}$, where \mathbf{g} is the acceleration due to gravity.

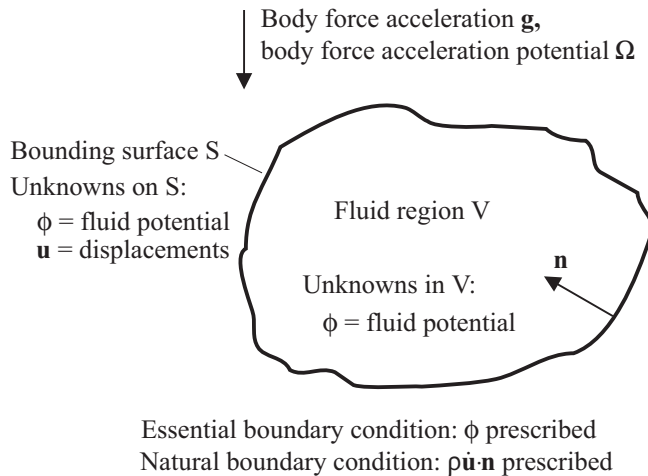


Figure 2.11-1: Fluid region

Equations (2.11-1) and (2.11-2) are valid for an inviscid irrotational fluid with no heat transfer. In particular, (2.11-2) is valid only when the pressure is a function of the density (and not of, for example, the density and temperature).

For the pressure-density relationship, we use the slightly compressible relationship

$$\frac{\rho}{\rho_0} = 1 + \frac{p}{\kappa} \quad (2.11-3)$$

where κ is the bulk modulus and ρ_0 is the nominal density. (2.11-3) then directly gives the density-enthalpy relationship and

the pressure-enthalpy relationships:

$$\rho = \rho_0 \exp\left(\frac{\rho_0 h}{\kappa}\right), p = \kappa \left[\exp\left(\frac{\rho_0 h}{\kappa}\right) - 1 \right] \quad (2.11-4a,b)$$

The continuity equation (2.11-1) is then approximated using the standard Galerkin approach to obtain

$$\delta F_\phi = \int_V (\dot{\rho} + \nabla \cdot (\rho \nabla \phi)) \delta \phi dV = 0 \quad (2.11-5)$$

where δ is the “variation of” symbol and δF_ϕ is the “variation in the mass flux rate”. (2.11-5) can be rewritten as

$$\delta F_\phi = \int_V (\dot{\rho} \delta \phi - \rho \nabla \phi \cdot \nabla \delta \phi) dV - \int_S \rho \nabla \phi \cdot \mathbf{n} \delta \phi dS \quad (2.11-6)$$

where S is the boundary of V and \mathbf{n} is the inwards normal on S . From this equation, we observe that the natural boundary condition is $\rho \nabla \phi \cdot \mathbf{n}$ prescribed, in other words, prescribed mass flux rate.

At this point, we notice that ρ is a function of the fluid velocity and position through the density-enthalpy relationship (2.11-4a), because the enthalpy is a function of fluid velocity and position. In V , we compute the enthalpy using (2.11.2). Hence, in V , the density is a function only of the fluid potential and position. But on S , we anticipate that part or all of S might be a moving boundary with velocity $\dot{\mathbf{u}}(\mathbf{x})$. We adopt the convention that the fluid velocity \mathbf{v} used in the enthalpy calculation on S is computed in terms of both the boundary velocity $\dot{\mathbf{u}}$ and the internal fluid velocity $\nabla \phi$ using

$$\mathbf{v}_n = (\dot{\mathbf{u}} \cdot \mathbf{n}) \mathbf{n}, \mathbf{v}_t = \nabla \phi - (\nabla \phi \cdot \mathbf{n}) \mathbf{n}, \mathbf{v} = \mathbf{v}_n + \mathbf{v}_t \quad (2.11-7a,b,c)$$

In other words, the fluid velocity normal to the surface is taken to be the velocity of the moving boundary and the remainder of the fluid velocity (which is tangential to the surface) is taken from the fluid potential. We also include the motions of S in the body

force potential calculations in the enthalpy. Combining the above considerations, we take

$$h = \Omega(\mathbf{x} + \mathbf{u}) - \dot{\phi} - \frac{1}{2} \mathbf{v}_n \cdot \mathbf{v}_n - \frac{1}{2} \mathbf{v}_t \cdot \mathbf{v}_t \quad (2.11-8)$$

on S .

(2.11-6) can be written as

$$\delta F_\phi = \int_V \left(\frac{\partial \rho}{\partial h} \dot{h} \delta \phi - \rho \nabla \phi \cdot \nabla \delta \phi \right) dV + \int_S -\rho \dot{\mathbf{u}} \cdot \mathbf{n} \delta \phi dS \quad (2.11-9)$$

The surface integral is obtained using (2.11-7).

On S , we assume that the boundary motion $\mathbf{u}(\mathbf{x})$ is small enough so that the volume V and the normal \mathbf{n} can be assumed constant. In other words, we perform all integrations on the undeformed fluid boundary. We discuss the implications of this assumption later.

- From (2.11-9), if no boundary conditions are applied to S , the boundary condition $\rho \dot{\mathbf{u}} \cdot \mathbf{n} = 0$ is implied. This boundary condition corresponds to no fluid flow through the boundary.

From (2.11-9), the natural boundary condition is $\rho \dot{\mathbf{u}} \cdot \mathbf{n} = \text{prescribed}$. We use this boundary condition to apply the structural motions to the fluid domain. We also use this boundary condition to model infinite fluid regions. Both of these cases are discussed in more detail below.

Notice that there is no boundary condition corresponding to flow tangential to the boundary. In other words, fluid can slip tangential to the boundary without restriction.

Also from (2.11-9), the essential boundary condition $\phi = \text{prescribed}$ is possible. Through (2.11-2) this boundary condition corresponds to a partially prescribed enthalpy and hence a partially prescribed pressure. The enthalpy and pressure are not fully prescribed since $\nabla \phi$ is not prescribed when ϕ is prescribed only on the boundary.

Modifications to equations of motion for the structure

- We assume that part of the boundary S is adjacent to the structure (Figure 2.11-2). The part of the boundary adjacent to the structure is denoted S_1 .

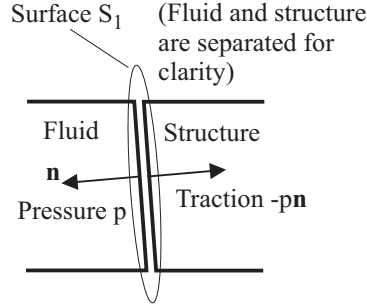


Figure 2.11-2: Forces on structure from fluid

The fluid pressure on S_1 provides additional forces on the structure adjacent to S_1 :

$$-\delta F_u = - \int_{S_1} p \mathbf{n} \cdot \delta \mathbf{u} dS_1 \quad (2.11-10)$$

where δF_u is the variation in the applied force vector and the minus signs are used in anticipation of F_u being considered an internal force vector in (2-11.12) below. The pressure is evaluated using

$$p = p(h) = p\left(\Omega(\mathbf{x} + \mathbf{u}) - \dot{\phi} - \frac{1}{2} \mathbf{v}_n \cdot \mathbf{v}_n - \frac{1}{2} \mathbf{v}_t \cdot \mathbf{v}_t\right) \quad (2.11-11)$$

We emphasize that \mathbf{n} points into the fluid (and out of the structure).

Finite element equations of motion

- Equations (2.11-6) and (2.11-9) are linearized using standard procedures to obtain

$$\begin{aligned}
& \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{M}_{FF} \end{bmatrix} \begin{bmatrix} \Delta \ddot{\mathbf{u}} \\ \Delta \ddot{\boldsymbol{\phi}} \end{bmatrix} + \begin{bmatrix} \mathbf{C}_{UU} & \mathbf{C}_{UF} \\ \mathbf{C}_{FU} & -(\mathbf{C}_{FF} + (\mathbf{C}_{FF})_S) \end{bmatrix} \begin{bmatrix} \Delta \dot{\mathbf{u}} \\ \Delta \dot{\boldsymbol{\phi}} \end{bmatrix} + \\
& \begin{bmatrix} \mathbf{K}_{UU} & \mathbf{K}_{UF} \\ \mathbf{K}_{FU} & -(\mathbf{K}_{FF} + (\mathbf{K}_{FF})_S) \end{bmatrix} \begin{bmatrix} \Delta \mathbf{u} \\ \Delta \boldsymbol{\phi} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \mathbf{F}_U \\ \mathbf{F}_F + (\mathbf{F}_F)_S \end{bmatrix}
\end{aligned} \tag{2.11-12}$$

where the increment in the vector of unknown potentials $\boldsymbol{\phi}$ is written $\Delta \boldsymbol{\phi}$ and the increment in the vector of unknown displacements \mathbf{u} is written $\Delta \mathbf{u}$. (We drop the left superscript $t + \Delta t$ here and below for ease of writing.) In the linearization process, increments in both the nodal displacements and increments in the nodal potentials (and their time derivatives) are considered. In (2-11.12),

\mathbf{F}_U = vector from (2.11-10)

\mathbf{F}_F = vector from volume integration term in (2.11-9)

$(\mathbf{F}_F)_S$ = vector from surface integration term in (2.11-9)

and the matrices are obtained by linearization. Note that vectors and matrices with the subscript S are integrated over the surface. Also the matrices with displacement degrees of freedom are integrated over the surface.

The sum of \mathbf{F}_F and $(\mathbf{F}_F)_S$ can be interpreted as an “out-of-balance” mass flux vector. (2-11.12) is satisfied only if this sum is zero at every node in the fluid, that is, if the consistent mass fluxes at each of the nodes from all of the attached fluid elements sum to zero.

In (2.11-12), we do not include any of the structural system matrices. (2.11-12) only gives the contribution of the potential-based fluid elements to the system matrices. Of course, the rest of the structure will make additional contributions to the first row of the above equations.

- Equation (2.11-12) is a nonlinear system of equations. The nonlinearities come from several sources:

- 1) The $-\frac{1}{2}\nabla\phi\cdot\nabla\phi$ term in h . This term is sometimes called the Bernoulli effect.
- 2) The nonlinear relationship between the density, pressure and the enthalpy (2.11-4).

Because the system is nonlinear, equilibrium iterations must be employed for an accurate solution, just as in nonlinear structural analysis.

- There is no explicit restriction on the magnitude of the fluid velocities in equation (2.11-12). However, in practice, there are several restrictions
 - 1) The velocity must be smaller than the speed of sound, otherwise, the equations (2.11-9) become hyperbolic and cannot be solved using ordinary finite element techniques.
 - 2) The density change should not be too great, otherwise the change in mass flux may be quite different than the change in volume flux when the boundary velocities are perturbed.
 - 3) Also, the pressure-density relationship in (2.11-3) only holds for relatively small density changes.

With these restrictions, the Mach number of the fluid flow should not exceed about 0.3 or so.

- Equation (2.11-12) is in general nonsymmetric. But in the limit of very small velocities, equation (2.11-12) becomes symmetric (see equation (2.11-19) below). Even for finite velocities, equation (2.11-12) is “nearly” symmetric. This means that we can symmetrize (2.11-12) and use the usual symmetric equation solvers of ADINA for the solution of (2.11-12). The convergence rate is very fast for small velocities and becomes slower as the velocities become larger.

2.11.2 Theory: Infinitesimal velocity formulation

- If we assume that the velocities and the density changes are infinitesimally small, then the continuity equation (2.11-1) becomes

$$\dot{\rho} + \nabla \cdot (\rho \nabla \phi) \approx \dot{\rho} + \rho_0 \nabla^2 \phi \approx \frac{\rho_0 \dot{p}}{\kappa} + \rho_0 \nabla^2 \phi = 0 \quad (2.11-13)$$

The momentum/equilibrium equation (2.11-2) becomes

$$h \approx \frac{p}{\rho} \approx \Omega(\mathbf{x}) - \dot{\phi} \quad (2.11-14)$$

from which we see

$$p \approx \rho (\Omega(\mathbf{x}) - \dot{\phi}) \approx \rho_0 (\Omega(\mathbf{x}) - \dot{\phi}) \quad (2.11-15)$$

Substituting (2.11-15) into (2.11-13) gives

$$-\rho_0 \ddot{\phi} + \kappa \nabla^2 \phi = -\rho_0 \dot{\Omega} \quad (2.11-16)$$

Equation (2.11-16) is a special form of the wave equation. It is linear in the solution variable ϕ . (2.11-16) can be written in variational form using standard techniques. The result is

$$\begin{aligned} & -\int_V \rho_0 \ddot{\phi} \delta\phi dV - \int_V \kappa \nabla \phi \cdot \nabla \delta\phi dV - \int_S \kappa \dot{\mathbf{u}} \cdot \mathbf{n} \delta\phi dS \\ & = -\int_V \rho_0 \dot{\Omega} \delta\phi dV \end{aligned} \quad (2.11-17)$$

The fluid pressure onto the structure becomes

$$\delta F_u = \int_{S_1} p \mathbf{n} \cdot \delta \mathbf{u} dS_1 \approx \int_{S_1} \left(\rho_0 \Omega + \rho_0 \frac{\partial \Omega}{\partial \mathbf{x}} \cdot \mathbf{u} - \rho_0 \dot{\phi} \right) \mathbf{n} \cdot \delta \mathbf{u} dS_1 \quad (2.11-18)$$

The finite element contributions to the system matrices corresponding to (2.11-17) and (2.11-18) are

$$\begin{aligned}
\begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{M}_{\text{FF}} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{U}} \\ \ddot{\boldsymbol{\phi}} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{C}_{\text{FU}}^T \\ \mathbf{C}_{\text{FU}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{U}} \\ \dot{\boldsymbol{\phi}} \end{bmatrix} + \begin{bmatrix} (\mathbf{K}_{\text{UU}})_S & \mathbf{0} \\ \mathbf{0} & -\mathbf{K}_{\text{FF}} \end{bmatrix} \begin{bmatrix} \mathbf{U} \\ \boldsymbol{\phi} \end{bmatrix} \\
= \begin{bmatrix} (\mathbf{R}_{\text{UB}})_S \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ -\dot{\mathbf{R}}_{\text{FB}} \end{bmatrix} \quad (2.11-19)
\end{aligned}$$

where

\mathbf{M}_{FF} = matrix from $\ddot{\boldsymbol{\phi}} \delta \boldsymbol{\phi}$ term in (2.11-17)

\mathbf{K}_{FF} = matrix from $\nabla \boldsymbol{\phi} \cdot \delta \nabla \boldsymbol{\phi}$ term in (2.11-17)

\mathbf{C}_{FU} = matrix from $\dot{\mathbf{u}} \cdot \mathbf{n} \delta \boldsymbol{\phi}$ term in (2.11-17)

$(\mathbf{K}_{\text{UU}})_S$ = matrix from $\left(\rho_0 \frac{\partial \Omega}{\partial \mathbf{x}} \cdot \mathbf{u} \right) \mathbf{n} \cdot \delta \mathbf{u}$ term in (2.11-18)

$(\mathbf{R}_{\text{UB}})_S$ = loads vector from $(\rho_0 \Omega) \mathbf{n} \cdot \delta \mathbf{u}$ term in (2.11-18)

$\dot{\mathbf{R}}_{\text{FB}}$ = loads vector from $\rho_0 \dot{\Omega} \delta \boldsymbol{\phi}$ term in (2.11-17)

\mathbf{U} = vector containing unknown nodal displacements

$\boldsymbol{\phi}$ = vector containing unknown nodal fluid potentials.

We note that the term $(\mathbf{K}_{\text{UU}})_S$ is numerically very small compared with the rest of the structural stiffness matrix, when there is a structure adjacent to the fluid. But $(\mathbf{K}_{\text{UU}})_S$ is important in the case when there is no structure adjacent to the fluid.

- The left-hand-side of equation (2.11-19), with the exception of the term $(\mathbf{K}_{\text{UU}})_S$, is identical to the formulation presented in the following reference:

ref. L.G. Olson and K.J. Bathe, "Analysis of fluid-structure interactions. A direct symmetric coupled formulation based on the fluid velocity potential", *J. Computers and Structures*, Vol 21, No. 1/2, pp 21-32, 1985.

2.11.3 Theory: Infinite fluid regions

- Both the subsonic and infinitesimal velocity formulations include special boundary conditions for the modeling of infinite fluid regions.
- The basic approach is to replace the infinite fluid region with a boundary condition that simulates the infinite fluid region. The boundary condition allows outwards-going waves to be propagated into the infinite fluid region without reflection.
- There are many approaches for setting up infinite elements. One approach that is physically intuitive is to use the results from acoustic analysis regarding outwards-going waves.

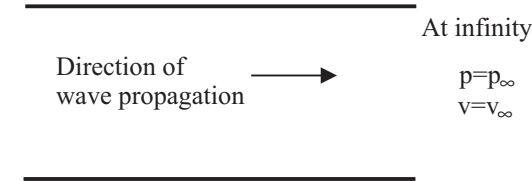
Planar waves: To illustrate the procedure, we consider a planar boundary, that is, a boundary on which plane acoustic waves propagate outwards through the boundary (Figure 2.11-3). If the waves have very small amplitude,

$$\Delta p = \rho c \Delta v \quad (2.11-20)$$

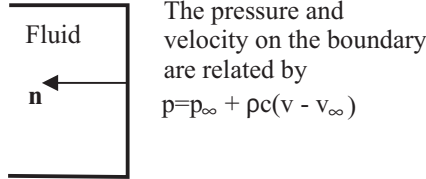
where Δp is the change in pressure applied to the boundary from the fluid on the outside of the boundary (in other words, the change in pressure applied to the boundary from the fluid that is not explicitly modeled), Δv is the change in the outwards velocity of the fluid and c is the speed of sound in the fluid. Hence the pressure on the boundary is

$$p = p_{\infty} + \rho c (v - v_{\infty}) \quad (2-11.21)$$

where v is the outwards velocity at the boundary, and p_{∞} and v_{∞} are the pressure and outwards velocity at infinity (p_{∞} and v_{∞} must be specified as part of the model definition).



a) Physical problem



b) Modeling with planar infinite boundary

Figure 2.11-3: Planar infinite boundary

The variation in mass flow through the boundary is

$$\delta F_{\phi_B} = - \int_{S_B} \rho \nabla \phi \cdot \mathbf{n} \delta \phi dS = \int_{S_B} \rho \mathbf{v} \delta \phi dS \quad (2.11-22)$$

We need to express the mass flow through the boundary in terms of the potential on the boundary. This can be done as follows. First, we assume that there is no fluid flow tangential to the boundary and that the boundary does not move. Under those conditions,

$$p(h) = p(\Omega - \dot{\phi} - \frac{1}{2} v^2) \quad (2.11-23)$$

Given a value of $\dot{\phi}$, v can be numerically evaluated by combining equations (2.11-21) and (2.11-23) and hence, given a value of $\dot{\phi}$, (2.11-22) can be numerically evaluated. We use this procedure in the implementation of the planar infinite element for the subsonic velocity formulation.

As an illustrative example, in the special case when there are no body forces, the velocities are small, and the velocities and pressure

at infinity are zero

$$\rho v = \frac{p}{c} = \frac{-\rho \dot{\phi}}{c} \quad (2.11-24)$$

so that the variation in mass flow rate is, in this special case,

$$\delta F_{\phi B} = - \int_{S_B} \frac{\rho \dot{\phi}}{c} \delta \phi dS \quad (2.11-25)$$

We use (2.11-25) in the implementation of the planar infinite element for the infinitesimal velocity formulation.

Spherical waves: The same technique can be used for spherical waves, provided that the appropriate relationship is used for the relationship between p and ρv . The relationship that we use for spherical waves is based on the relationship between pressure and mass flux in the frequency domain

$$\rho v = \frac{p}{c} + \frac{p}{i\omega r} \quad (2.11-26)$$

where $i = \sqrt{-1}$, ω is the frequency of the outwards-going wave and r is the radius of the boundary. Here we assume that both the pressure and velocity at infinity are equal to zero and that the velocity itself is small. The equivalent of (2.11-26) in the time domain is

$$\rho v = \frac{p}{c} + \frac{\int p dt}{r} \quad (2.11-27)$$

We now neglect body force effects, so that $p = -\rho \dot{\phi}$ on the boundary. Then

$$\rho v = \frac{-\rho \dot{\phi}}{c} - \frac{\rho \dot{\phi}}{r} \quad (2.11-28)$$

so that the variation in mass flow rate is

$$\delta F_{\phi B} = - \int_{S_B} \left(\frac{\rho \dot{\phi}}{c} + \frac{\rho \phi}{r} \right) \delta \phi dS \quad (2.11-29)$$

Note that this derivation holds regardless of the wave frequency ω

Cylindrical waves: The same technique and derivation can be used for cylindrical waves. For cylindrical waves, the relationship between pressure and mass flux in the frequency domain is

$$\rho v = \frac{p}{ic} \left[\frac{J_1(kr) - iY_1(kr)}{J_0(kr) - iY_0(kr)} \right] \quad (2.11-30)$$

where J_0, J_1, Y_0, Y_1 are the Bessel's functions of the first and second kinds of orders 0 and 1, and $kr = \frac{\omega r}{c}$. (2.11-30) cannot be transformed into the time domain, but for frequencies higher than about $\frac{\omega r}{c} = 1$, it turns out that (2.11-30) is well approximated by (2.11-26), so that the spherical wave derivation can be used, provided that the frequency is high enough.

- For planar, spherical and cylindrical infinite boundaries, only surface integrations are required. Also, no additional degrees of freedom are required on the boundary. For planar waves, the infinite boundaries contribute to $(\mathbf{F}_F)_S$ $(\mathbf{C}_{FF})_S$ and for spherical and cylindrical waves, the infinite boundaries contribute to $(\mathbf{F}_F)_S$, $(\mathbf{C}_{FF})_S$ and $(\mathbf{K}_{FF})_S$.
- It is instructive to compare the above approach with the doubly-asymptotic approach in Olson and Bathe.

ref. L.G. Olson and K.J. Bathe, "An infinite element for analysis of transient fluid-structure interactions", *Engineering Computations*, Vol 2, pp 319-329, 1985.

In the DAA, the infinite fluid region is modeled as the superposition of two effects, an added mass approximation and a plane wave approximation. The added mass approximation dominates for low frequencies and the plane wave approximation dominates for high frequencies. The plane wave approximation gives exactly the results given above for planar waves (equation 2.11-25). But the added mass approximation in the DAA requires a volume integral and hence volume infinite elements are required for modeling non-planar waves. Provided that the nodes in the volume infinite elements “at infinity” are placed at the correct locations, the volume infinite elements give exactly the same results as the spherical infinite elements (equation (2.11-29).

Since the surface integral formulation given in equations (2.11-20) to (2.11-30) gives the same results as the DAA approach, and since the surface integral formulation does not require volume elements, we feel that the surface integral formulation is preferable.

2.11.4 Theory: Ground motion loadings

- Ground motion loadings require special treatment. In this case,

$$\mathbf{U} = \mathbf{U}_r + \mathbf{U}_g \quad (2.11-31)$$

where \mathbf{U}_g is the vector of nodal point ground displacements and \mathbf{U}_r is the vector of nodal point displacements relative to the ground motion. The ground motions are expressed as

$$\mathbf{U}_g = \sum_k u_{gk} \mathbf{d}_k \quad (2.11-32)$$

where u_{gk} are the ground displacements in direction k and \mathbf{d}_k is the vector of nodal point values in which $\mathbf{d}_k(i) = 1$ if equation i corresponds to a translation in direction k and $\mathbf{d}_k(i) = 0$ otherwise.

Because the ground motions are known, the increment in displacements $\Delta \mathbf{U}$ is equal to the increment in the relative displacements $\Delta \mathbf{U}_r$. Hence the left hand sides of equations (2.11-12) and (2.11-19) are unchanged and the right-hand-sides of these

equations are updated only by additional internal forces. For example, the additional internal forces added to (2.11-19) from the fluid elements are

$$-\begin{bmatrix} (\mathbf{K}_{UU})_S \mathbf{d}_k \\ \mathbf{0} \end{bmatrix} u_{gk} - \begin{bmatrix} \mathbf{0} \\ -\mathbf{C}_{FU} \mathbf{d}_k \end{bmatrix} \dot{u}_{gk}$$

Note that ϕ still contains the nodal point values of ϕ and ϕ is the potential corresponding to the absolute (not relative) velocities.

Also note that ground motions are handled differently than physical body forces. This is quite different than in ordinary structural analysis, in which physical body forces and ground motions are both modeled using mass-proportional loads.

2.11.5 Theory: Static analysis

- The case of static analysis also requires special attention. It is not correct simply to set the velocities to zero in (2.11-1), because (2.11-1) is identically satisfied for zero velocities.
- We envision the process of obtaining a static solution as a quasi-static process. In this process, all fluid velocities are assumed negligible and therefore the fluid potential is constant (in space). Under these conditions,

$$h = \Omega - \dot{\phi} \quad (2.11-32)$$

and the variational statement of continuity is

$$\delta F_\phi = \int_V \dot{\rho} \delta \phi dV - \int_S \rho \dot{\mathbf{u}} \cdot \mathbf{n} \delta \phi dS \quad (2.11-33)$$

Equation (2.11-33) is simply the integral equation of conservation of mass because $\delta \phi$ is constant in space.

- We now numerically integrate (2.11-33) in time. We use the Euler backwards approximations

$${}^{t+\Delta t} \dot{\rho} = ({}^{t+\Delta t} \rho - {}^t \rho) / \Delta t, \quad {}^{t+\Delta t} \dot{\mathbf{u}} = ({}^{t+\Delta t} \mathbf{u} - {}^t \mathbf{u}) / \Delta t \quad (2.11-34a,b)$$

and then obtain, at time $t + \Delta t$,

$$\delta F_\phi = \int_V \left({}^{t+\Delta t}\rho - {}^t\rho \right) \delta \dot{\phi} dV - \int_S {}^{t+\Delta t}\rho \left({}^{t+\Delta t}\mathbf{u} - {}^t\mathbf{u} \right) \cdot \mathbf{n} \delta \dot{\phi} dS \quad (2.11-35)$$

where we write $\delta \dot{\phi} = \delta \phi / \Delta t$.

The modification to the structural equations of motion is

$$\delta F_u = \int_{S_1} {}^{t+\Delta t}p \mathbf{n} \cdot \delta \mathbf{u} dS_1 \quad (2.11-36)$$

which is the same as equation (2.11-10), except that the $t + \Delta t$ is explicitly written.

We prefer the Euler backwards approximations because then the stiffness matrix remains symmetric when the displacement increments are infinitesimally small. Of course, the steps must be small enough so that the Euler backwards approximations apply, but in many problems, these approximations are quite good, especially when the fluid is almost incompressible.

Note that we assume that the domain of integration and its boundary remain (nearly) unchanged during the time integration. All integrations are performed over the original domain of integration.

Also note that the solution in the fluid is the single value $\dot{\phi}$. This value represents the Bernoulli constant.

The linearized equations of motion for the subsonic formulation become

$$\begin{bmatrix} \left(\tilde{\mathbf{K}}_{UU} \right)_S & \tilde{\mathbf{K}}_{UF} \\ \tilde{\mathbf{K}}_{FU} & -\left(\tilde{\mathbf{K}}_{FF} + \left(\tilde{\mathbf{K}}_{FF} \right)_S \right) \end{bmatrix} \begin{bmatrix} \Delta \mathbf{u} \\ \Delta \dot{\phi} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \tilde{\mathbf{F}}_U \\ \tilde{\mathbf{F}}_F + \left(\tilde{\mathbf{F}}_F \right)_S \end{bmatrix} \quad (2.11-37)$$

where we use the \sim to emphasize that the vectors and matrices are different than the corresponding dynamic vectors and matrices, along with the constraints $\dot{\phi} = \text{constant}$ (if there is more than one fluid region in the problem, then $\dot{\phi} = \text{constant}$ in each fluid

region). As in the dynamic case (equation 2.11-12), equation (2.11-37) is in general non-symmetric, but (2.11-37) becomes symmetric when the displacement increment ${}^{t+\Delta t}\mathbf{u} - {}^t\mathbf{u}$ becomes infinitesimally small. So, as in the dynamic case, we symmetrize the system and use the usual symmetric equation solver.

In (2.11-37) the sum of $\tilde{\mathbf{F}}_F + (\tilde{\mathbf{F}}_F)_S$ can be interpreted as an “out-of-balance” mass vector. (2.11-37) is only satisfied if the total mass of each fluid region is conserved. Motions of the boundary that do not change the total mass of any fluid region are associated with zero pivots in the system matrices, unless there is structural stiffness associated with these motions. Note that the $(\tilde{\mathbf{K}}_{UU})_S$ matrix also provides structural stiffness, so if there are body forces, motions of the boundary that do not change the total mass of any fluid region are not associated with zero pivots.

The equations of motion for the infinitesimal velocity formulation can be obtained either from the above derivation, or can be formally derived from (2.11-19) by applying the Laplace transform to both sides of (2.11-19) and applying the final value theorem. The result is

$$\begin{bmatrix} (\mathbf{K}_{UU})_S & \mathbf{C}_{FU}^T \\ \mathbf{C}_{FU} & -\mathbf{M}_{FF} \end{bmatrix} \begin{bmatrix} \mathbf{U} \\ \dot{\boldsymbol{\phi}} \end{bmatrix} = \begin{bmatrix} (\mathbf{R}_{UB})_S \\ -\mathbf{R}_{FB} \end{bmatrix} \quad (2.11-38)$$

together with the condition

$$\mathbf{K}_{FF}\boldsymbol{\phi} = \mathbf{0} \quad (2.11-39)$$

- There are a number of unusual characteristics of (2.11-37), (2.11-38) and (2.11-39):
 - The solution involves $\dot{\boldsymbol{\phi}}$ instead of $\boldsymbol{\phi}$. This makes sense as (2.11-32) then implies that p is constant (in time) in a static solution.
 - The condition (2.11-39) must be satisfied. When there are no infinite boundaries, this condition is satisfied whenever

$\dot{\phi} = \text{constant}$ within each separate fluid region. Hence the number of unknown potential degrees of freedom in static analysis is equal to the number of separate fluid regions in the analysis.

- The condition $\dot{\phi} = \text{constant}$ within each separate fluid region implies that $p = \rho_0 \Omega + C$ where C is a constant determined from the solution. Hence the variation of pressure within each separate fluid region is contained within C and any choice of constant of integration within $\rho_0 \Omega$ (recall that Ω is a potential and therefore includes an arbitrary constant of integration) is balanced by an opposite change in C .
- If the ϕ degree of freedom is fixed at a node, or if infinite boundaries are included in the fluid region, then you must set $\phi = 0$ for all the nodes in the fluid region. This condition implies $p = \rho_0 \Omega$ within the fluid region. Any choice of constant of integration within Ω then affects the solution.
- It is necessary to enter the density of the fluid in static analysis, even when the solution does not depend on the density.

2.11.6 Theory: Frequency analysis

- Frequency analysis is possible when there is no structural damping, when the infinitesimal velocity formulation is used, and when there are no infinite boundaries. The eigenvalue problem to be solved is

$$\left(-\omega_j^2 \begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_{\text{FF}} \end{bmatrix} - \omega_j \begin{bmatrix} \mathbf{0} & \mathbf{C}_{\text{FU}}^T \\ \mathbf{C}_{\text{FU}} & \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{K} + (\mathbf{K}_{\text{UU}})_s & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{\text{FF}} \end{bmatrix} \right) \begin{bmatrix} \mathbf{U}^{(j)} \\ \mathbf{F}^{(j)} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix} \quad (2.11-40)$$

where $\mathbf{F}^{(j)} = -i \dot{\phi}^{(j)}$, $i = \sqrt{-1}$ and in which we also include the structural stiffness matrix \mathbf{K} and structural mass matrix \mathbf{M} .

(2.11-40) is derived from (2.11-19) by taking the Fourier transform of the left-hand-side.

(2.11-40) is a non-standard eigenvalue problem in which all of the eigenvalues are real and non-negative. It is solved using either the determinant search method or the Lanczos iteration method.

The eigenvectors are scaled according to the following orthogonality condition:

$$\begin{aligned} \mathbf{u}^{(i)} \mathbf{C}_{\text{FU}}^T \mathbf{F}^{(i)} + \mathbf{F}^{(i)} \mathbf{C}_{\text{FU}} \mathbf{u}^{(i)} + (\omega_i + \omega_j) (\mathbf{u}^{(i)} \mathbf{M}_{\text{UU}} \mathbf{u}^{(i)} + \mathbf{F}^{(i)} \mathbf{M}_{\text{FF}} \mathbf{F}^{(i)}) \\ = 2\sqrt{\omega_i \omega_j} \delta_{ij} \end{aligned} \quad (2.11-41)$$

where δ_{ij} is the Kronecker delta. Notice that when there is no fluid, (2.11-41) reduces to the usual orthogonality condition $\mathbf{u}^{(i)} \mathbf{M}_{\text{UU}} \mathbf{u}^{(i)} = \delta_{ij}$.

(2.11-40) has one rigid body mode for each separate fluid region in which all of the ϕ degrees of freedom are free. Each rigid body mode is of the form $\mathbf{U}^{(i)} = 0$, $\mathbf{F}^{(i)} = \text{constant}$ in each separate fluid region. For future reference, we term these rigid body modes ϕ rigid body modes. The determinant search method can determine all ϕ rigid body modes. The Lanczos frequency solver does not print the ϕ rigid body modes.

The modal pressures and modal fluid particle displacements are computed using

$$p^{(j)} = \rho \omega_j f^{(j)}, \quad \mathbf{u}^{(j)} = \frac{1}{\omega_j} \nabla f^{(j)} \quad (2.11-42)$$

in which $p^{(j)}$ is the modal pressure at the point of interest for mode j , $f^{(j)}$ is the fluid potential eigenvector $\mathbf{F}^{(j)}$ interpolated to the point of interest and $\mathbf{u}^{(j)}$ is the modal displacement vector at the point of interest for mode j .

2.11.7 Theory: Mode superposition

- Mode superposition is possible when there is no structural

damping, the infinitesimal velocity formulation is used and when there are no infinite boundaries. The theory used to determine the modal equations of motion is given in Chapter 6 of the following reference:

ref. L. Meirovitch, *Computational Methods in Structural Dynamics*, Sijthoff & Noordhoff, 1980.

The derivation starts with the equations of motion (2.11-19) being put into the form

$$\begin{aligned} \begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_{FF} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{U}}_r \\ \ddot{\boldsymbol{\phi}} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{C}_{FU}^T \\ -\mathbf{C}_{FU} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{U}}_r \\ \dot{\boldsymbol{\phi}} \end{bmatrix} + \begin{bmatrix} \mathbf{K} + (\mathbf{K}_{UU})_S & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{FF} \end{bmatrix} \begin{bmatrix} \mathbf{U}_r \\ \boldsymbol{\phi} \end{bmatrix} \\ = \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} (\mathbf{R}_{UB})_S \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \dot{\mathbf{R}}_{FB} \end{bmatrix} - \begin{bmatrix} (\mathbf{K}_{UU})_S \mathbf{d}_k \\ \mathbf{0} \end{bmatrix} u_{gk} + \begin{bmatrix} \mathbf{0} \\ \mathbf{C}_{FU} \mathbf{d}_k \end{bmatrix} \dot{u}_{gk} \end{aligned} \quad (2.11-43)$$

where we include the structural stiffness matrix \mathbf{K} and structural mass matrix \mathbf{M} . We also include the ground motion loading terms in (2.11-43) in anticipation of the response spectrum derivation below. (2.11-43) corresponds to an undamped gyroscopic system. The modal expansion used is

$$\mathbf{U}_r = \sum_{j=1}^n \frac{\xi_j}{\omega_j} \mathbf{U}^{(j)}, \quad \dot{\boldsymbol{\phi}} = \sum_{j=1}^n -\xi_j \mathbf{F}^{(j)} \quad (2.11-44a,b)$$

where ξ_j is the generalized coordinate for mode j . The modal equation is

$$\ddot{\xi}_j + \omega_j^2 \xi_j = \Gamma^{(j)} \quad (2.11-45)$$

where

$$\Gamma^{(j)} = -(\mathbf{F}^{(j)})^T \dot{\mathbf{R}}_\phi + \omega_j (\mathbf{U}^{(j)})^T \mathbf{R}_u \quad (2.11-46)$$

and, from (2.11-43),

$$\begin{aligned}\mathbf{R}_u &= \mathbf{R} + (\mathbf{R}_{UB})_S - (\mathbf{K}_{UU})_S \mathbf{d}_k u_{gk} \\ \mathbf{R}_\phi &= \mathbf{C}_{FU} \mathbf{d}_k \dot{u}_{gk}\end{aligned}\quad (2.11-47a,b)$$

With modal damping (damping ratio ζ_j), (2.11-45) becomes

$$\ddot{\xi}_j + 2\omega_j \zeta_j \dot{\xi}_j + \omega_j^2 \xi_j = \Gamma^{(j)} \quad (2.11-48)$$

Then we use

$$\mathbf{U}_r = \sum_{j=1}^n \frac{\xi_j}{\omega_j} \mathbf{U}^{(j)}, \quad \dot{\mathbf{U}}_r = \sum_{j=1}^n \frac{\dot{\xi}_j}{\omega_j} \mathbf{U}^{(j)}, \quad \ddot{\mathbf{U}}_r = \sum_{j=1}^n \frac{\ddot{\xi}_j}{\omega_j} \mathbf{U}^{(j)} \quad (2.11-49a,b,c)$$

$$\phi = \sum_{j=1}^n \int_0^t -\xi_j dt \mathbf{F}^{(j)}, \quad \dot{\phi} = \sum_{j=1}^n -\xi_j \mathbf{F}^{(j)}, \quad \ddot{\phi} = \sum_{j=1}^n -\dot{\xi}_j \mathbf{F}^{(j)} \quad (2.11-50a,b,c)$$

to obtain the structural and fluid response.

The initial modal coordinates are computed from the initial conditions (displacements, velocities, fluid potentials and time derivatives of fluid potentials). Initial accelerations and initial second time derivatives of fluid potentials are not used.

In the first solution step, we use the Newmark method with $\alpha = 1/2$, $\delta = 1$ because this choice of Newmark parameters does not require initial accelerations. In the successive solution steps, we use the Newmark method with the usual Newmark parameters $\alpha = 1/4$, $\delta = 1/2$.

2.11.8 Theory: Response spectrum, harmonic and random vibration analysis

- Modal participation factors are calculated for response spectrum analysis, harmonic vibration analysis and random vibration analysis. There are two cases: modal participation factors corresponding to applied forces (in which case we do not consider

response spectrum analysis) and modal participation factors corresponding to ground motions.

In the modal participation factor calculation, we only consider the dynamic solution (due to dynamically applied loads).

To begin the derivation, we introduce

$$x_j = \frac{\xi_j}{\omega_j} \quad (2.11-51)$$

as the new generalized coordinate. The modal expansion is then

$$\mathbf{U}_r = \sum_{j=1}^n x_j \mathbf{U}^{(j)}, \quad \dot{\boldsymbol{\phi}} = \sum_{j=1}^n -\omega_j x_j \mathbf{F}^{(j)} \quad (2.11-52a,b)$$

and the modal equation of motion is

$$\ddot{x}_j + 2\omega_j \zeta_j \dot{x}_j + \omega_j^2 x_j = -\frac{1}{\omega_j} (\mathbf{F}^{(j)})^T \dot{\mathbf{R}}_\phi + (\mathbf{U}^{(j)})^T \mathbf{R}_u \quad (2.11-53)$$

Applied forces: As it is assumed that the physical body forces are entirely static and that there are no ground motions, the modal equation of motion becomes

$$\ddot{x}_j + 2\omega_j \zeta_j \dot{x}_j + \omega_j^2 x_j = (\mathbf{U}^{(j)})^T \mathbf{R} \quad (2.11-54)$$

and evidently the modal participation factor is simply

$$\bar{F}^{(j)} = (\mathbf{U}^{(j)})^T \mathbf{R} \quad (2.11-55)$$

We use $\bar{F}^{(j)}$ rather than the $F^{(j)}$ given in (2.11-45) because then the modal expansion for \mathbf{U}_r is then the commonly used one.

As a consequence of (2.11-55), the modal participation factor for any ϕ rigid body mode is zero.

Ground motions: The loads vectors are

$$\begin{aligned}\mathbf{R}_u &= -(\mathbf{K}_{UU})_S \mathbf{d}_k u_{gk} - \mathbf{M} \mathbf{d}_k \ddot{u}_{gk} , \\ \mathbf{R}_\phi &= \mathbf{C}_{FU} \mathbf{d}_k \dot{u}_{gk}\end{aligned}\quad (2.11-56a,b)$$

Hence

$$\begin{aligned}\ddot{x}_j + 2\omega_j \zeta_j \dot{x}_j + \omega_j^2 x_j &= - \left(\frac{1}{\omega_j} (\mathbf{F}^{(j)})^T \mathbf{C}_{FU} \mathbf{d}_k + (\mathbf{U}^{(j)})^T \mathbf{M} \mathbf{d}_k \right) \ddot{u}_{gk} \\ &\quad - (\mathbf{U}^{(j)})^T (\mathbf{K}_{UU})_S \mathbf{d}_k u_{gk}\end{aligned}\quad (2.11-57)$$

We neglect the term $(\mathbf{U}^{(j)})^T (\mathbf{K}_{UU})_S \mathbf{d}_k u_{gk}$ (which is probably small anyway) to obtain

$$\ddot{x}_j + 2\omega_j \zeta_j \dot{x}_j + \omega_j^2 x_j = -\bar{F}^{(j)} \ddot{u}_{gk} \quad (2.11-58)$$

where the ground motion modal participation factor is

$$\bar{F}^{(j)} = \frac{1}{\omega_j} (\mathbf{F}^{(j)})^T \mathbf{C}_{FU} \mathbf{d}_k + (\mathbf{U}^{(j)})^T \mathbf{M} \mathbf{d}_k$$

This factor has the usual physical interpretation, provided that each fluid region is completely surrounded by interface elements.

The modal participation factor for a ϕ rigid body mode can be zero or non-zero. It can be shown that the modal participation factor for a ϕ rigid body mode is zero if each fluid region is completely surrounded by interface elements.

Note: static corrections (residual calculations) are not implemented for the potential-based fluid element because residual displacement calculations are based upon non-constant (in space) body force loading. Such loading is not possible in general for the potential-based fluid elements.

2.11.9 Modeling: Formulation choice and potential master degree of freedom

The potential-based formulation discussed in this section is introduced in ADINA 7.4 and extensively modified in ADINA 8.0.

You can select the potential-based formulation in the AUI command-line input using the parameter FLUIDPOTENTIAL in the MASTER command:

FLUIDPOTENTIAL=AUTOMATIC: The potential-based formulation of ADINA 7.4 and 8.0 is employed. This is the default.

FLUIDPOTENTIAL=YES: The potential-based formulation of ADINA 7.3 and lower is employed.

FLUIDPOTENTIAL=NO: The potential-based formulation is not employed.

Note, it is not possible to select the potential-based formulation using the AUI user interfaces. The potential-based formulation of ADINA 8.0 is employed when using the AUI user interfaces.

When FLUIDPOTENTIAL=AUTOMATIC, the AUI automatically detects the presence of potential-based elements, and, if there are any potential-based elements, activates the potential master degree of freedom.

2.11.10 Modeling: Potential degree of freedom fixities

Subsonic formulation: Deleting the potential degree of freedom along part of the bounding surface has the effect of partially specifying the enthalpy along that surface, see equation (2.11-2). This has no physical meaning.

Infinitesimal velocity formulation: Deleting the potential degree of freedom along part of the bounding surface has the effect of setting the pressure equal to $\rho_0 \Omega$ along that part of the bounding surface, see equation (2.11-15). If there are no body forces, then $\Omega = 0$ and the pressure is therefore set to zero along that part of the bounding surface. This boundary condition can be used if the displacements of the boundary are not of interest.

It is recommended that the potential degrees of freedom not be deleted anywhere in the fluid.

2.11.11 Modeling: Elements

The volume V of the fluid domain is modeled using two-dimensional or three-dimensional fluid elements. These elements are analogous to the two-dimensional or three-dimensional solid elements and the nodal point numbering of the fluid elements is the same as the nodal point numbering of the solid elements.

The two-dimensional elements are either planar (unit thickness of fluid assumed) or axisymmetric (1 radian of fluid assumed).

These elements are defined within 2-D and 3-D fluid element groups (in ADINA). Set the formulation of the element group to either “Linear Potential-Based Element” (for the infinitesimal velocity formulation) or “Subsonic Potential-Based Element” (for the subsonic velocity formulation).

The bounding surface S of the fluid domain is modeled with potential-interfaces and/or interface elements, as discussed in detail below.

It is required that each separate fluid domain be modeled with separate fluid element groups. This is because the AUI constrains the potential degrees of freedom of each element group together in static analysis during phi model completion, step 7, see Section 2.11.15.

It is not permitted to have fluid regions of different densities sharing the same potential degrees of freedom. This is because the nodal pressure would be different as computed from the fluid regions connected to the node.

It is recommended that the fluid and adjacent structure *not* share the coincident nodes. This allows the AUI to construct appropriate constraint equations between the fluid and structural degrees of freedom that are most appropriate during phi model completion, see Section 2.11.15.

2.11.12 Modeling: Potential-interfaces

For ease of modeling, you can define potential-interfaces of various types along the surface of the fluid domain. When you generate a data file, the AUI places interface elements along the boundaries specified by the potential-interfaces, except as noted below. ADINA itself does not use the potential-interfaces.

There are several types of potential-interface:

Fluid-structure: Place a fluid-structure potential interface on the boundary between a potential-based fluid and the adjacent structure.

In many cases, the AUI can automatically generate fluid-structure interface elements along the boundary between the fluid and structure during phi model completion, step 1, see Section 2.11.15. So you typically do not need to define fluid-structure potential-interfaces.

Free surface: Place a free surface potential interface on the boundary where the pressures are to be prescribed and the displacements are desired, for example, on the free surface of a fluid. Surface waves can be approximately modeled in this manner, but note that the displacement of the waves is assumed to be small.

ADINA-F: Place an ADINA-F potential interface on the boundary adjacent to an ADINA-F mesh.

Usually the AUI automatically generates ADINA-F interface elements along the boundary adjacent to an ADINA-F mesh during phi model completion, step 1. So you typically do not need to define ADINA-F potential-interfaces.

Infinite: Place an infinite potential-interface wherever infinite boundary conditions are desired.

There are three types of infinite potential-interface.

Planar: In the subsonic formulation, the pressure and velocity “at infinity” must be specified. In the infinitesimal velocity formulation, the pressure and velocity are assumed to be zero.

Spherical: The radius of the boundary must be specified. The pressure and velocity at infinity are assumed to be zero, and the velocities at the boundary are assumed to be small.

Cylindrical: The radius of the boundary must be specified. The pressure and velocity at infinity are assumed to be zero, and the velocities at the boundary are assumed to be small. This element cannot accurately model low-frequency waves, that is,

waves with $\frac{\omega r}{c} < 1$.

Inlet-outlet: Place an inlet-outlet potential-interface wherever the pressure of the boundary is specified, and where the displacements of the boundary are not of interest. For example, the outlet of a pipe on which the pressure is known can be modeled using an inlet-outlet potential-interface.

If there is more than one inlet or outlet in the model, each inlet and outlet must have its own inlet-outlet potential-interface.

Fluid-fluid: Place a fluid-fluid potential-interface on the boundary between two potential-based fluid elements of two different element groups.

Note that only one fluid-fluid potential-interface need be defined for each boundary. The AUI generates a fluid-fluid interface element for each of the two elements that share a common boundary during phi model completion, step 1, see Section 2.11.15.

Rigid-wall: Place a rigid-wall potential interface wherever the fluid is not to flow through the boundary.

Note that this boundary condition is modeled in ADINA by the absence of any interface element. Therefore a rigid-wall potential-interface suppresses any automatic generation of interface elements along the boundary of the rigid-wall potential-interface. However the AUI uses the rigid-wall potential interface during phi model completion, step 2, in constructing structural normals, see Section 2.11.15.

2.11.13 Modeling: Interface elements

Interface elements are used on the surface of the fluid domain to specify a boundary condition. In many cases, you do not have to define interface elements. Rather you define potential-interfaces and the AUI then generates the interface elements. However we describe the interface elements here for completeness.

Interface elements are defined within the same element group as the fluid elements themselves. In 2-D analysis, the fluid-structure interface elements are 2 or 3 node line segments, in 3-D analysis, the fluid-structure interface elements are 3 to 9 node area segments. The nodal point numbering conventions are shown in Figure

2.11-4. The AUI automatically reorders all interface elements when generating the ADINA input data file so that the interface element normals point into the fluid.

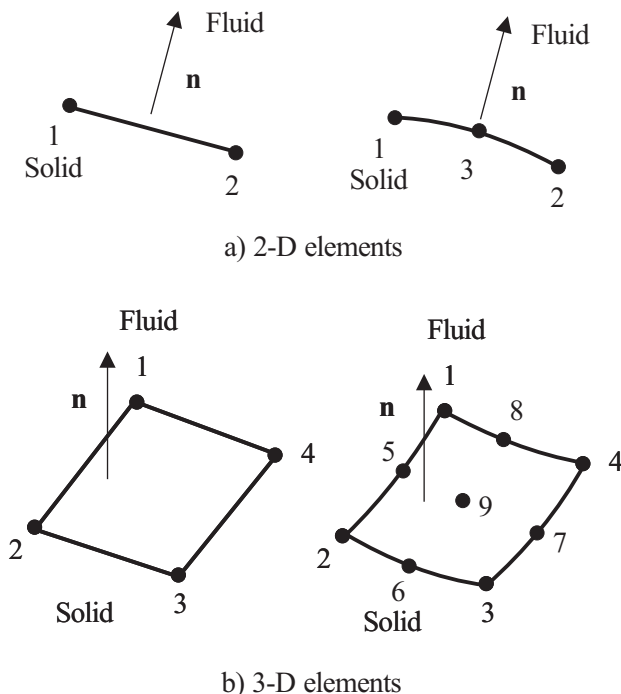


Figure 2.11-4: Fluid-structure interface elements, showing the local node numbering convention

There are several types of interface elements:

Fluid-structure interface element: This element connects the potential-based fluid element with an adjacent structural element (Figure 2.11-5). Each node of the element contains the potential degree of freedom and displacement degrees of freedom. It is assumed that the displacements of the nodes of the interface element are small.

It is assumed that the structure provides stiffness to all translational degrees of freedom, because the fluid-structure interface element does not provide stiffness, mass or damping to the tangential directions.

We emphasize that the potential-based fluid element, fluid-

structure interface element and adjacent structural element must all be compatible.

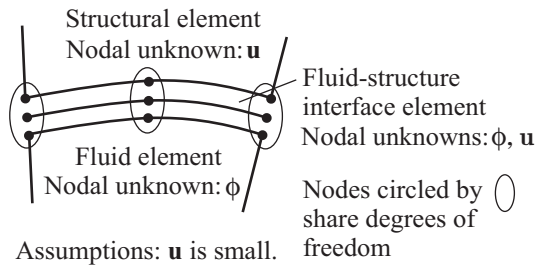


Figure 2.11-5: Fluid-structure interface element

Free surface interface element: This element is placed onto the boundary of a potential-based fluid element wherever the pressure is to be prescribed, and wherever the displacements of the fluid are required (Figure 2.11-6). For example, the free surface of a fluid in a basin can be modeled using free surface interface elements. Each node of the element contains the potential degree of freedom and displacement degrees of freedom. It is assumed that displacements and velocities of the nodes of the interface element are small.

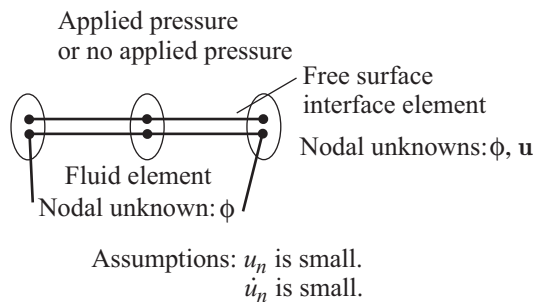


Figure 2.11-6: Free surface interface element

It is necessary to fix all displacements that are tangential to the free surface interface element, because the free surface interface element does not provide stiffness, mass or damping to the tangential directions.

In many cases, the AUI can generate skew systems and fixities corresponding to the tangential directions during phi model

generation, see Section 2.11.15.

ADINA-F interface element: This element is placed onto the boundary of a potential-based fluid element wherever the boundary is adjacent to an ADINA-F mesh (Figure 2.11-7). Each node of the element contains the potential degree of freedom and displacement degrees of freedom. It is assumed that displacements and velocities of the nodes of the interface element are small.

Note that the coupling between ADINA and ADINA-F is through the displacements of the shared boundary. Therefore, because the displacements of the ADINA-F interface element are assumed to be small, the ADINA/ADINA-F coupling cannot be used for modeling actual fluid flow between the ADINA and ADINA-F models.

It is necessary to fix all displacements that are tangential to the ADINA-F interface element, because the ADINA-F interface element does not provide stiffness, mass or damping to the tangential directions. In many cases, the AUI can generate these fixities automatically during phi model completion, see Section 2.11.15.

The ADINA-F interface element need not be compatible with the elements from the adjacent ADINA-F mesh.

It is necessary to define an ADINA-F fluid-structure boundary in the ADINA model to connect the ADINA and ADINA-F models, just as in ordinary ADINA-F FSI analysis.

In most cases, the AUI automatically generates the ADINA-F interface elements during phi model completion, step 1, see Section 2.11.15.

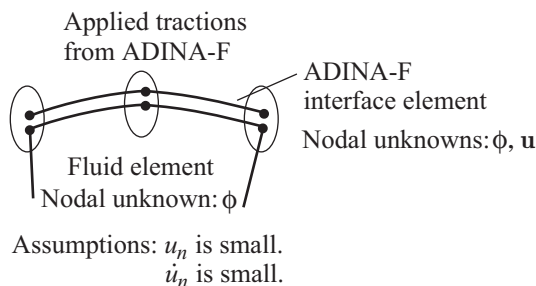


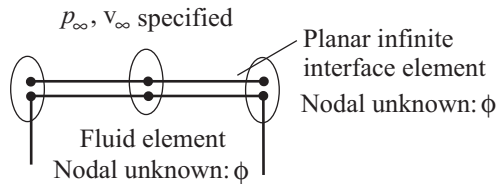
Figure 2.11-7: ADINA-F interface element

Infinite interface elements: This element is placed onto the

boundary of a potential-based fluid element wherever infinite boundary conditions are desired. Each node of the element contains only a potential degree of freedom.

There are three types of infinite interface element:

Planar infinite element: In the subsonic formulation, the pressure and velocity “at infinity” must be specified. In the infinitesimal velocity formulation, the pressure and velocity are assumed to be zero. A planar infinite element is shown in Figure 2.11-8.



Assumptions: ϕ constant along boundary.

Flow velocity is close to v_∞

Figure 2.11-8: Planar infinite interface element

Spherical infinite element: The radius of the boundary must be specified. The pressure and velocity at infinity are assumed to be zero, and the velocities at the boundary are assumed to be small.

Cylindrical infinite element: The radius of the boundary must be specified. The pressure and velocity at infinity are assumed to be zero, and the velocities at the boundary are assumed to be small. This element cannot accurately model low-frequency waves, that is, waves with $\frac{\omega r}{c} < 1$.

When considering where to place the infinite interface elements, remember that the infinite elements are derived under the assumption that waves travel normal to the boundary. Hence the boundary must be placed so that any anticipated waves travel normal to the boundary (Figure 2.11-9).

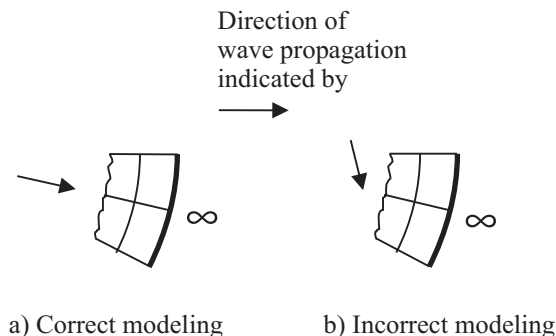


Figure 2.11-9: Direction of wave propagation must be normal to the infinite interface element

Inlet-outlet interface element: This element is placed onto the boundary of a potential-based fluid element wherever the pressure of the boundary is specified, and where the displacements of the boundary are not of interest (Figure 2.11-10). For example, the outlet of a pipe on which the pressure is known can be modeled using inlet-outlet interface elements. Each node of the element contains the potential degree of freedom and displacement degrees of freedom. The displacement degrees of freedom are only used to compute velocities and accelerations; the displacements themselves are not used.

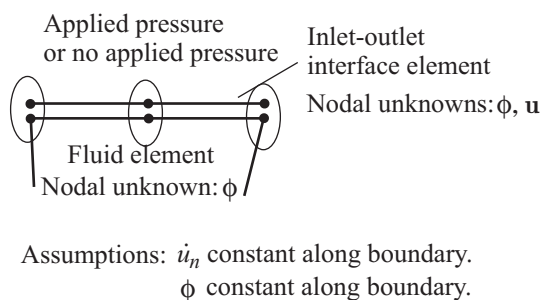


Figure 2.11-10: Inlet-outlet interface element

It is necessary to fix all displacements that are tangential to the inlet-outlet interface element, because the inlet-outlet interface element does not provide stiffness, mass or damping to the tangential directions. In addition, it is necessary to set the tangential fluid velocity to zero by constraining all of the potential

degrees of freedom to be equal. Finally it is necessary to set the normal velocity to be uniform along the element, by constraining the normal velocities to be equal. In many cases, the AUI can generate these fixities and constraints automatically during phi model completion, see Section 2.11.15.

Fluid-fluid interface element: This element is placed along the interface between two potential-based fluid elements of two different element groups (Figure 2.11-11). For example, the interface between air and water can be modeled using fluid-fluid interface elements.

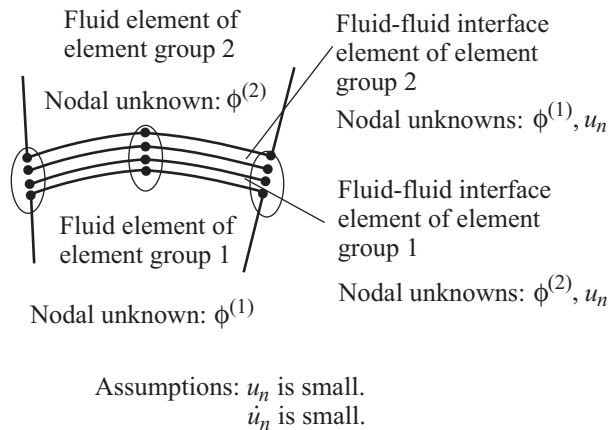


Figure 2.11-11: Fluid-fluid interface element

Each node of the interface element contains the potential degree of freedom and displacement degrees of freedom. The normal velocities and displacements are assumed to be small. It is necessary to fix all displacements that are tangential to the fluid-fluid interface element, because the fluid-fluid interface element does not provide stiffness, mass or damping to the tangential directions.

Each of the potential-based elements that share a common boundary requires a fluid-fluid interface element. The two fluids are connected by constraining the normal displacements of one of the interface elements to the corresponding normal displacements of the other interface element.

In many cases, the AUI can generate these fixities and constraints automatically during phi model completion, see Section 2.11.15.

2.11.14 Modeling: Loads

Concentrated forces, pressure loads, prescribed displacements

Concentrated forces, pressure loads and/or prescribed displacements can be applied directly to any part of the fluid boundary on which there are fluid-structure, free surface, inlet-outlet or fluid-fluid interface elements. However, when applying concentrated forces, remember that the AUI can apply skew systems to certain nodes on the fluid boundary during phi model completion, see Section 2.11.15. Therefore you should make sure that the nodes on which you apply concentrated forces have the degree of freedom directions that you anticipate.

Mass-proportional loads

Mass-proportional loads can be applied. However the AUI and ADINA make a distinction between those mass-proportional loads used to model physical body forces and those mass-proportional loads used to enter ground accelerations. For each mass-proportional load, you must specify its interpretation: body force or ground acceleration.

Body force: Mass-proportional loads interpreted as physical body forces must be constant in time when there are infinitesimal velocity potential-based fluid elements in the model. These loads are used in the construction of g_k and Ω as follows:

$$g_k = \text{MAGNITUDE} \times A(k) \times \text{TF}$$

where MAGNITUDE is the magnitude of the mass-proportional loading, $A(k)$ is the vector giving the direction of the mass-proportional loading and TF is the value of the time function. Then

$$\Omega = \sum_{k=1}^3 g_k (x_k - x_{k0}) \quad (2.11-59)$$

where x_{k0} is a datum value (entered as part of the fluid material description). Notice that the choice of the datum value affects the solution only when infinite interface elements are present, or if at least one potential degree of freedom is fixed in the fluid region.

Body forces can be applied in a static analysis as the only loads in the analysis. Then, if other time-varying loads are present, a restart to dynamic analysis can be performed. The body forces should be kept in the restart dynamic analysis.

Ground acceleration: Mass-proportional loads interpreted as ground motions can be time-varying, and are used in the construction of u_{gk} as follows: Suppose that you specify a mass-proportional load interpreted as ground motions of magnitude $g_k(t)$, where $g_k(t)$ is entered as for body force loads (but the time function need not be constant). Then ADINA computes the ground motions using

$$\ddot{u}_{gk} = -g_k, \quad \dot{u}_{gk} = \int_{tstart}^t \ddot{u}_{gk} dt, \quad u_{gk} = \int_{tstart}^t \dot{u}_{gk} dt \quad (2.11-60a,b,c)$$

Note that it is assumed that $\dot{u}_{gk}(tstart) = 0$, $u_{gk}(tstart) = 0$.

For example, if $g_k = \sin \omega t$, then the above equations imply

$$\dot{u}_{gk} = -\frac{1}{\omega}(1 - \cos \omega t), \quad u_{gk} = -\frac{1}{\omega} \left(t - \frac{\sin \omega t}{\omega} \right).$$

It is seen that the average ground velocity is non-zero, which is probably not realistic. As an alternative, use

$$\begin{aligned} g_k &= \frac{t}{t_p} \sin \omega t, & t < t_p, \\ &= \sin \omega t, & t \geq t_p \end{aligned}$$

in which $t_p = \frac{2\pi}{\omega}$ is one period of the ground motion.

The choice of interpretation for the mass-proportional loads does not affect the structural elements used in the model.

Centrifugal loads

Centrifugal load effects are not included in the potential-based fluid elements.

Mass flux loads (phiflux loads)

Mass flux loads (also referred to as phiflux loads) can be prescribed directly onto potential-based fluid elements. The mass fluxes can be distributed or concentrated. The dimensions of

distributed mass flux are $\frac{[\text{mass}]}{[\text{time}] \times [\text{area}]}$ in dynamic analysis and

are $\frac{[\text{mass}]}{[\text{area}]}$ in static analysis). The dimensions of concentrated

mass flux are $\frac{[\text{mass}]}{[\text{time}]}$ in dynamic analysis and are $[\text{mass}]$ in static analysis.

Positive mass flux is assumed to represent mass flowing into the fluid domain.

No interface elements or potential-interfaces should be defined on boundaries with distributed or concentrated mass fluxes.

2.11.15 Modeling: Phi model completion

As can be seen above, there are many restrictions and conditions that must be considered when specifying boundary conditions on potential-based fluid elements. Many of these conditions have been automated in the AUI in the following way. The AUI performs “phi model completion” whenever generating a data file in which potential-based fluid elements are used. The steps in phi model completion are:

1) The AUI loops over all fluid element sides on the boundary of each fluid element region. If the fluid element side already has an interface element, the side is skipped. If the fluid element side has a potential-interface, an interface element of the appropriate type is generated. Otherwise, the side is checked to see if it is attached to a structure (shares structural degrees of freedom with structural elements), close to a structure (nodes coincident with nodes of a structural element) or on a ADINA-F fluid-structure boundary; and, if any of the above conditions are met, an interface element of the appropriate type is generated. The intent of this step is to cover as much of the fluid boundary as possible with interface elements.

Note that phi model completion relies on determining the nodes that are “close” together. The tolerances used in this determination are controlled by command PHI-MODEL-COMPLETION, parameters CLOSE-TOL, XTOL, YTOL, ZTOL. By default, the tolerances used during phi model completion are the same tolerances used in mesh generation coincident node checking .

2) The AUI loops over all nodes attached to interface elements. If the node is attached to structural elements, the node is skipped. Otherwise the types of the attached interface elements are determined. Then

a) If the node is attached only to a free surface interface, ADINA-F interface, inlet-outlet interface or fluid-fluid interface, then the node has a free normal direction (normal to the interface) and zero stiffness directions that are tangential to the free normal. The free normal and zero stiffness directions are identified, and if they are not aligned with the global directions, a skew system is generated that is aligned with the free normal and zero stiffness directions.

b) If the node is attached to a free surface interface, ADINA-F interface, inlet-outlet interface or fluid-fluid interface, and is also attached to a fluid-structure interface or rigid-wall interface, the AUI proceeds as follows. The node has a free normal direction (determined from the free surface interface, ADINA-F interface, inlet-outlet interface or fluid-fluid interface), a structural normal direction (determined from the fluid-structure interface or rigid-wall interface), and, in 3D, another direction orthogonal to the free

normal and structural normal directions, which may be a zero stiffness direction or another structural normal direction. The free normal direction is modified to be orthogonal to the structural normal directions. The free normal, structural normal and zero stiffness directions are identified, and, if they are not aligned with the global directions, a skew system is generated that is aligned with the free normal, structural normal and zero stiffness directions.

This process relies on an angle tolerance in 3D analysis to determine if the structure is “smooth”. This tolerance is parameter PHI-ANGLE in command PHI-MODEL-COMPLETION. See Example 2 below.

The intent of step 2 is to identify the zero stiffness and free normal directions of the nodes.

3) The AUI loops over all nodes attached to interface elements. If the node is attached to a structural element, the node is skipped. If the node (node A) is attached to a fluid-structure interface element and is close to a structural node B, node A is constrained to node B as follows. Each displacement degree of freedom for node A is constrained to the corresponding degrees of freedom for node B, accounting for differences in skew systems between A and B, accounting for B possibly being a slave node in a constraint equation (but not accounting for B possibly being a slave in a rigid link), accounting for B possibly being fixed. But a displacement degree of freedom for node A is not constrained if the degree of freedom is a free normal direction (see 2 above).

The intent of step 3) is to connect the fluid mesh with the structural mesh, when different nodes are used for the fluid and structure. The connection still allows the fluid nodes to slip relative to the structural nodes on intersections between free surfaces and the structure.

4) In static analysis, when there are no body force loads, the AUI loops over all nodes on a free surface, ADINA-F interface, or fluid-fluid interface. If the node is attached to a structural element, the node is skipped. Otherwise, constraint equations are defined for all nodes so that the displacements in the direction of the free normal are equal.

The intent is to remove the zero pivots in the system matrices that are otherwise present (see the discussion after equation (2.11-37)).

5) The AUI loops over all nodes on an inlet-outlet interface. If the node is attached to a structural element, the node is skipped. Otherwise, constraint equations are defined for all nodes so that the displacements in the direction of the free normal are equal.

The intent is to make the displacement / velocity constant over the inlet-outlet interface.

6) In dynamic analysis, or in static analysis when there are body force loads, the AUI loops over all nodes on a fluid-fluid interface. If the node is attached to a structural element, the node is skipped. Otherwise constraint equations are defined for all pairs of nodes, so that the displacements in the direction of the free normal are compatible.

The intent is to enforce displacement compatibility between the fluids.

7) The AUI then loops over all nodes with zero stiffness degrees of freedom and defines fixities for each zero stiffness degree of freedom.

8) In static analysis, the AUI constrains all of the potential degrees of freedom for an element group together.

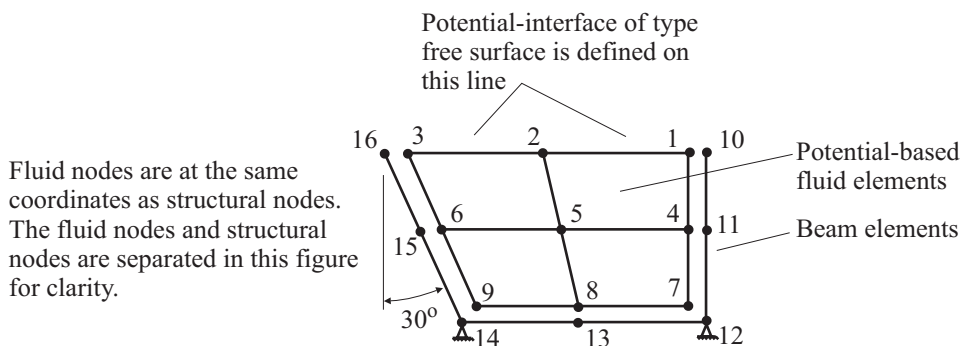
9) In dynamic analysis, the AUI constrains all of the potential degrees of freedom on an inlet-outlet together. The intent is to remove the tangential flows on an inlet-outlet.

Note that the model that results from the phi model completion process is not stored in the AUI, but is immediately written to the ADINA data file. Therefore the model that results from phi model completion cannot be displayed during model definition. Phi model completion does write some messages to the log file or user interface, which you may find helpful. However, we recommend that you also check the model by running the model through ADINA, for example, for one load step; then use ADINA-PLOT to display the model.

Example 1: We now present a detailed example for a 2-D fluid filled basin with flexible walls.

Figure 2.11-12(a) shows the model before phi model completion. The model is defined with separate nodes for the fluid and the structure. One way to guarantee that the nodes on the fluid and structure are separate is to generate the structural elements as usual, but to set coincidence checking to “group” when generating the fluid elements.

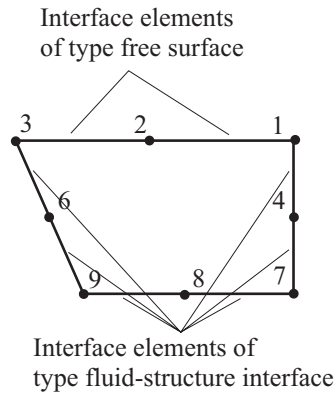
A potential-interface of type free surface is defined on the geometry line corresponding to the free surface.



a) Finite element model before phi model completion

Figure 2.11-12: Example 1 of phi model completion

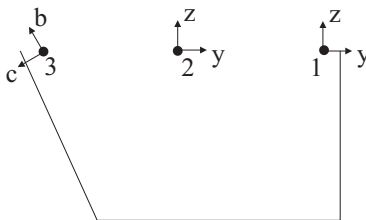
In step 1 of phi model completion, the AUI generates fluid-structure interface elements where the fluid is adjacent to the structure, and free surface interface elements corresponding to the potential-interface (Figure 2.11-12(b)).



b) Step 1 of phi model completion;
interface elements are created

Figure 2.11-12: (continued)

In step 2 of phi model completion, the AUI classifies the displacement directions on the free surface (Figure 2.11-12(c)). Notice that the free normal for node 2 is taken from the free surface, but the free normal for node 3 is modified by the presence of the adjacent structure. A skew system is defined for node 3 because the free normal and structural normal are not aligned with the global coordinate directions. The zero stiffness direction of node 2 will be fixed in step 6 below.



Node 1: y = structural normal direction,
z = free normal direction

Node 2: y = zero stiffness direction,
z = free normal direction

Node 3: b = free normal direction,
c = structural normal direction

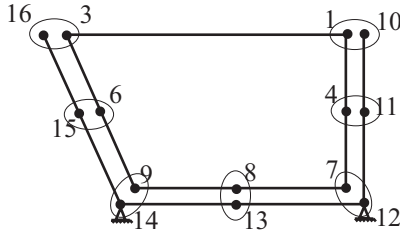
c) Step 2 of phi model completion: classification
of displacement directions on free surface

Figure 2.11-12: (continued)

In step 3 of phi model completion, the AUI constrains the fluid displacement directions to the structure (Figure 2.11-12(d)). At node 1, only the y displacement direction is constrained; the z displacement is left free so that the free surface can slip along the wall. At node 3, only the c displacement direction is constrained; the b displacement is left free so that the free surface can slip along the wall.

Notice that at node 4, both the y and z displacements are constrained to the structure. The fluid still slips in the z direction because only the normal displacement (the y displacement in this case) is used by the fluid equations. Similar statements hold for nodes 6 and 8.

Nodes 7 and 9 are fixed because corresponding nodes 12 and 14 are fixed.



Node 1: $u_y = u_y^{10}$

Node 3: $u_c = -\cos 30^\circ u_y^{16} - \sin 30^\circ u_z^{16}$

Node 4: $u_y = u_y^{11}, u_z = u_z^{11}$ Node 6: $u_y = u_y^{15}, u_z = u_z^{15}$

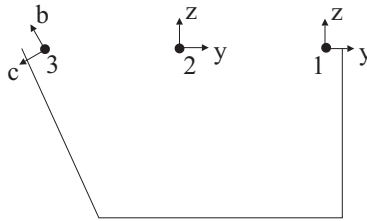
Node 7: $u_y = u_z = \text{fixed}$ Node 8: $u_y = u_y^{13}, u_z = u_z^{13}$

Node 9: $u_y = u_z = \text{fixed}$

d) Step 3: Creation of constraint equations and fixities

Figure 2.11-12: (continued)

If the analysis is static without body forces, then the AUI performs step 4 of phi model completion (Figure 2.11-12(e)). The free surface can only translate vertically as a rigid body.



Node 2: $u_z = u_z^1$

Node 3: $u_b = u_z^1 / \cos 30^\circ$

Step 4 is only performed in static analysis when there are no body forces.

e) Step 4 of phi model completion: defining constraint equations to set normal displacements equal on free surface

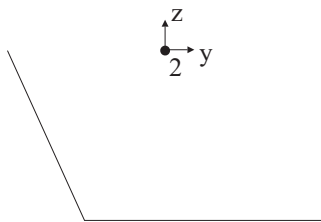
Figure 2.11-12: (continued)

This motion affects the total mass of the fluid region, so that there is no zero pivot in the system matrices.

If there are body forces, then step 4 is not necessary because all boundary motions are given stiffness by the matrix $(\mathbf{K}_{UU})_S$.

Step 5 of phi model completion is skipped because there are no fluid-fluid interfaces.

In step 6 of phi model completion, the zero stiffness direction at node 2 is fixed (Figure 2.11-12(f)). Vertical motions of the nodes attached only to free surface interface elements are allowed, but horizontal motions of these nodes are not allowed (because the fluid does not provide stiffness, damping or mass to horizontal motions).

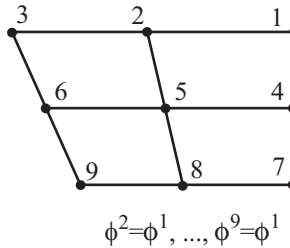


Node 2: u_y =fixed

f) Step 6 of phi model completion: defining fixities to eliminate zero stiffness degrees of freedom

Figure 2.11-12: (continued)

If the analysis is static, then the AUI performs step 7 of phi model completion (Figure 2.11-12g). Only constant (in space) potentials are allowed in static analysis.



Step 7 is performed only in static analysis

g) Step 7: defining constraint equations to set all potential degrees of freedom equal

Figure 2.11-12: (continued)

Example 2: An alternative way to model the problem of Example 1 is shown in Figure 2.11-13. Here the free surface is shifted downwards slightly so that the nodal coincidence checking feature will not reuse any existing nodes on the free surface boundary. Depending on how much you shift the fluid free surface, you may need to adjust the tolerance used in the coincidence checking algorithms.

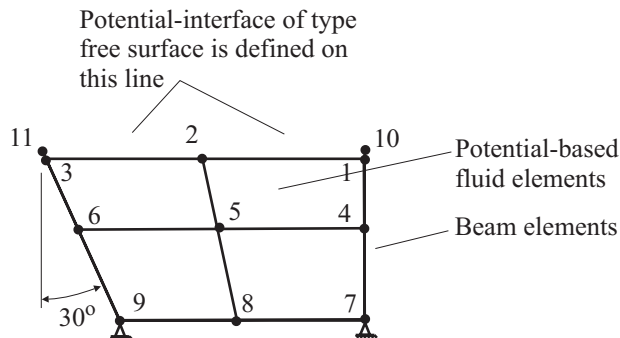


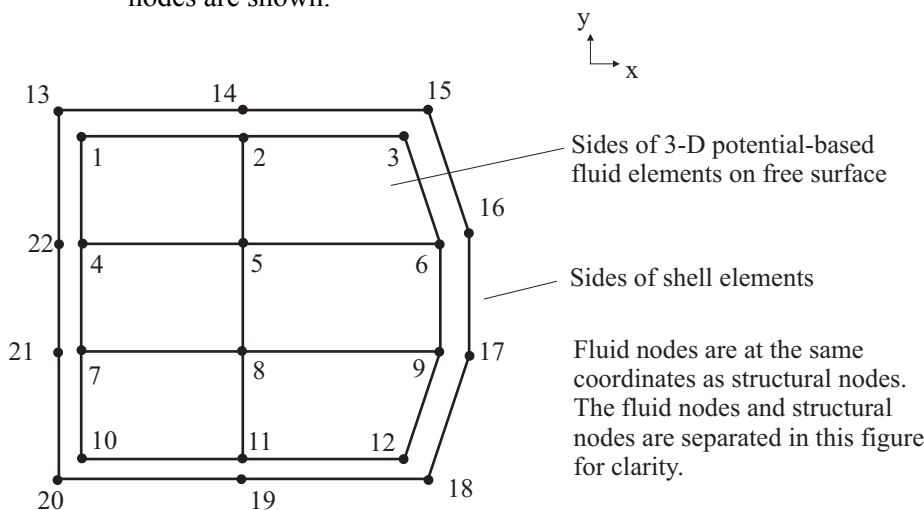
Figure 2.11-13: Alternative modeling of problem in Figure 2.11-12

During phi model completion, it is necessary to set the coincidence tolerance to be loose enough so that node 1 is considered “close” to node 10 and node 3 is considered to be close to node 11. Then steps 1, 2, 4, 6, 7 of phi model completion proceed as in Example 1, and step 3 only processes nodes 1 and 3. The solution results will be almost exactly the same as in Example

1 (the solutions will be slightly different because the geometry is slightly different).

We think that the user input for Example 2 is more difficult than the user input for Example 1 because Example 2 requires a good working knowledge of the nodal coincidence checking features.

Example 3: In 3-D analysis of a fluid-filled basin, there is one additional consideration. Consider the model shown in Figure 2.11-14, in which only the free surface and the adjacent structural nodes are shown.

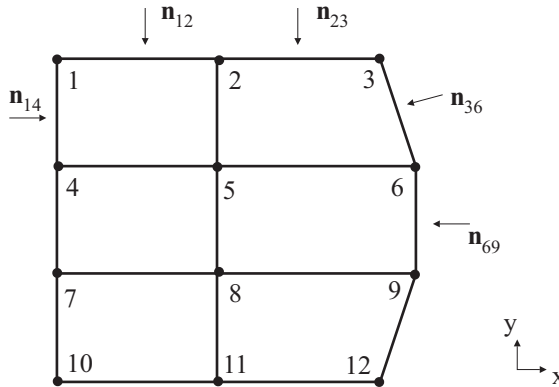


a) Top view of finite element model

Figure 2.11-14: Example 3 of phi model completion

During step 2 of phi model completion, the AUI determines the structural normal direction(s), zero stiffness direction(s) and free normal direction for the nodes of the free surface (Figure 2.11-14(b)). Notice that node 3 has two structural normals, but node 6 has only one structural normal. That is because the angle between the two structural normals for node 3 is greater than PHI-ANGLE, but the angle between the two structural normals for node 6 is less than PHI-ANGLE.

Nodes 3, 6, 9 and 12 are assigned skew systems because the structural normal directions are not aligned with the global system.



Node 1: Structural normal 1 = \mathbf{n}_{14}
 Structural normal 2 = \mathbf{n}_{12}
 Free normal = \mathbf{z}

Node 2: Structural normal = $\mathbf{n}_{12} = \mathbf{n}_{23}$
 Zero stiffness direction = \mathbf{x}
 Free normal = \mathbf{z}

Node 3: Structural normal 1 = \mathbf{n}_{23}
 Structural normal 2 = \mathbf{n}_{36}
 Free normal = \mathbf{z}

Node 5: Zero stiffness direction 1 = \mathbf{x}
 Zero stiffness direction 2 = \mathbf{y}
 Free normal = \mathbf{z}

Node 6: Structural normal = average of \mathbf{n}_{36} and \mathbf{n}_{69}
 Free normal = \mathbf{z}
 Zero stiffness direction = remaining orthogonal direction

b) Classification of structural normals and zero stiffness directions for some nodes on the free surface

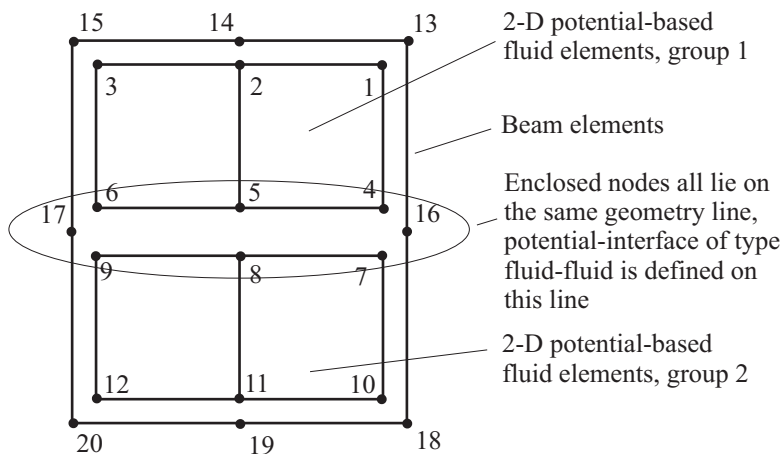
Figure 2.11-14: (continued)

During step 3 of phi model completion, the AUI creates constraint equations for the fluid nodes adjacent to the structural nodes. For example, node 1 is constrained in both the x and y directions to node 13, because both directions are structural normal directions. Node 2 is also constrained in both the x and y directions to node 14, here because the y direction is a structural normal direction and the x direction is a zero stiffness direction. (It is assumed that the structure provides stiffness in the x direction.)

During step 6 of phi model completion, the AUI fixes the x and y directions for nodes 5 and 8, because these directions are zero stiffness directions, and there is no adjacent structure.

Example 4: We now present some of the steps for phi model completion of an enclosure with two distinct fluid regions. Figure 2.11-15(a) shows the model before phi model completion.

Fluid nodes are at the same coordinates as structural nodes.
The fluid nodes and structural nodes are separated in this figure for clarity.



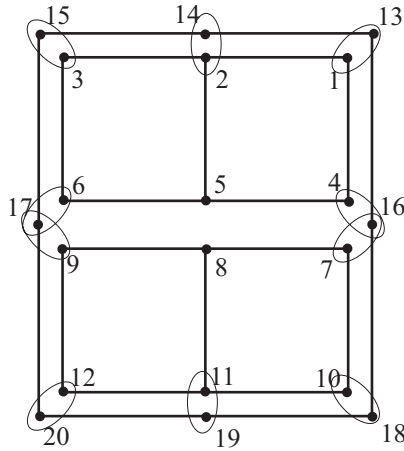
a) Finite element model before phi model completion

Figure 2.11-15: Example 4 of phi model completion

In step 1 of phi model completion, the AUI generates fluid-structure interface elements where the fluid is adjacent to the structure, and fluid-fluid interface elements corresponding to the potential-interface. Four fluid-fluid interface elements are generated, two for each shared element side.

In step 2 of phi model completion, the AUI classifies the displacement directions on the fluid-fluid interface. Here the free normal is always in the z direction and the zero stiffness directions for nodes 5 and 8 are in the y direction.

In step 3 of phi model completion, the fluid nodes are constrained to the adjacent structural nodes (Figure 2.11-15(b)). Notice that the nodes on the fluid-fluid interface are allowed to slip relative to the structure.



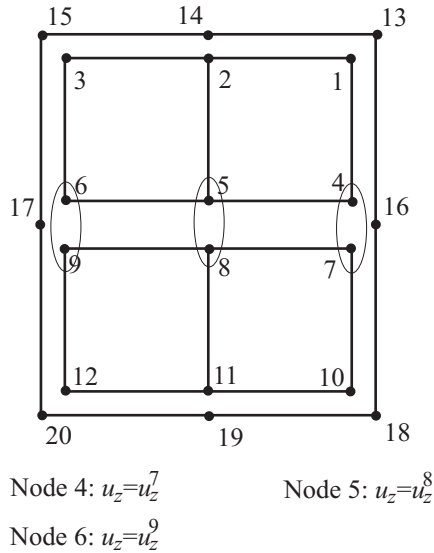
Node 1: $u_y = u_y^{13}, u_z = u_z^{13}$ Node 2: $u_y = u_y^{14}, u_z = u_z^{14}$
Node 3: $u_y = u_y^{15}, u_z = u_z^{15}$ Node 4: $u_y = u_y^{16}$
Node 6: $u_y = u_y^{17}$ Node 7: $u_y = u_y^{16}$
Node 9: $u_y = u_y^{17}$ Node 10: $u_y = u_y^{18}, u_z = u_z^{18}$
Node 11: $u_y = u_y^{19}, u_z = u_z^{19}$ Node 12: $u_y = u_y^{20}, u_z = u_z^{20}$

b) Step 3: constraining fluid nodes to adjacent structural nodes

Figure 2.11-15: (continued)

If the analysis is static without body forces, then the AUI performs step 4 of phi model completion. In this case, the z displacements of nodes 5 to 9 are constrained to be equal to the z displacement of node 4. The free surface can only translate vertically as a rigid body.

In step 5 of phi model completion, the fluid nodes on the fluid-fluid interface are constrained to each other (Figure 2.11-15(c)). Step 5 is not performed if step 4 was performed. Notice that the potential degrees of freedom are not constrained.

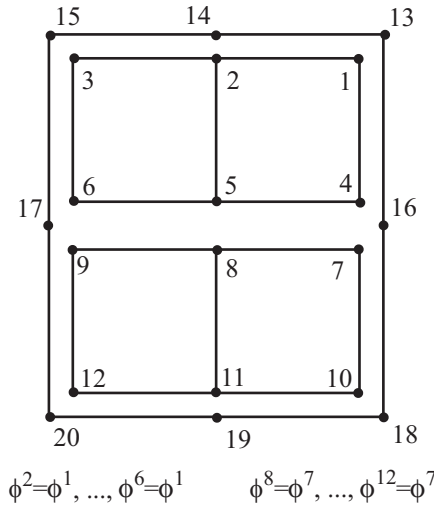


c) Step 5: constraining adjacent nodes of the two fluid groups together

Figure 2.11-15: (continued)

In step 6 of phi model completion, the zero stiffness direction of nodes 5 and 8 are fixed. Vertical motions of the nodes attached only to fluid-fluid interface elements are allowed, but horizontal motions are not allowed (because the fluid does not provide stiffness, damping or mass to horizontal motions).

If the analysis is static, then the AUI performs step 7 of phi model completion (Figure 2.11-15(d)). Only constant (in space) potentials are allowed in static analysis, but the potential can be different for the two fluid element groups.



Step 7 is performed only in static analysis

- d) Step 7: defining constraint equations to set the potential degrees of freedom of each element group together

Figure 2.11-15: (continued)

Example 5: We demonstrate how to model an initial pressure in a confined fluid region. This can be done in a static analysis as follows: Define an auxiliary fluid element, physically unattached to the fluid region but belonging to the same fluid element group. On the auxiliary fluid element, place a potential-interface of type inlet-outlet, and apply the specified initial pressure to the inlet-outlet.

The constraint equations that the AUI defines in static analysis connects the auxiliary fluid element to the confined fluid region. The result is that the value of $\dot{\phi}$ in the confined fluid region is the same as the value of $\dot{\phi}$ in the auxiliary fluid element; the value of $\dot{\phi}$ in the auxiliary fluid element is determined by the pressure that you apply to the inlet-outlet.

If you restart to a dynamic or frequency analysis, fix all of the degrees of freedom of the auxiliary fluid element. Since, in dynamic or frequency analysis, the AUI does not define constraint equations connecting the potential degrees of freedom, the auxiliary fluid element is no longer connected to the confined fluid

region. The initial values of $\dot{\phi}$ in the confined fluid region are that from the static analysis, so the confined fluid region has the same initial pressure as in static analysis.

The modeling process is schematically illustrated in Figure 2.11-16.

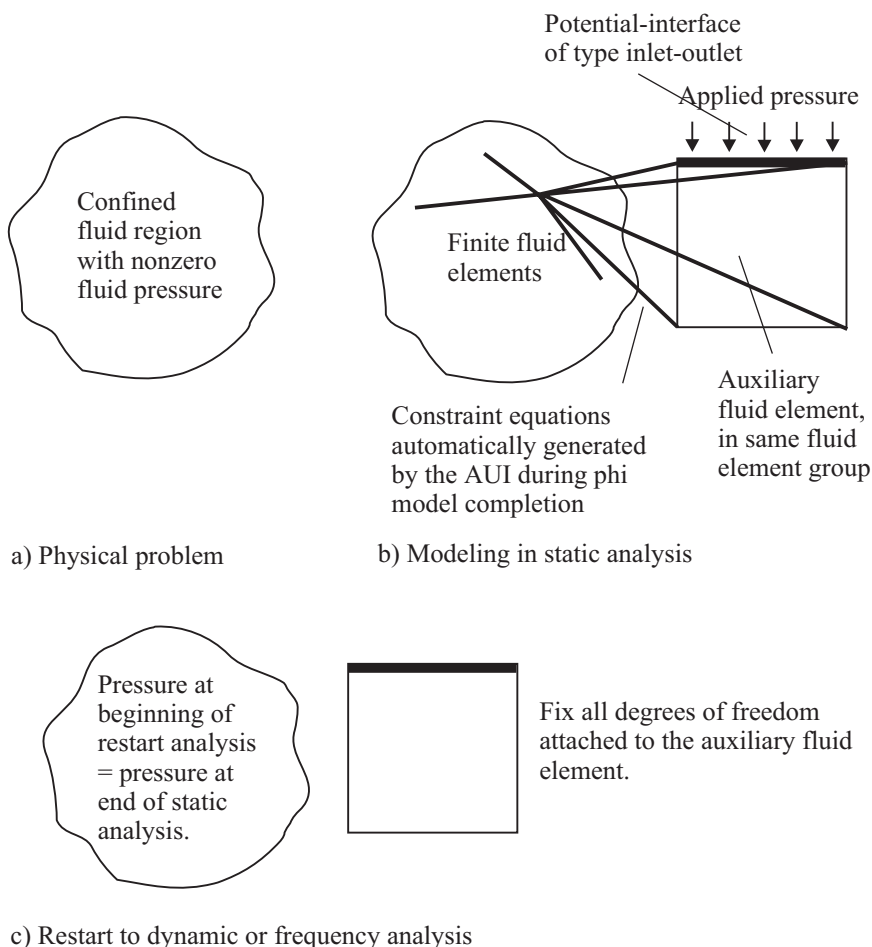


Figure 2.11-16: Modeling an internal pressure in a confined fluid region

2.11.16 Modeling: Considerations for static analysis

In static analysis, the potential degree of freedom output should be interpreted as $\dot{\phi}$, not ϕ .

2.11.17 Modeling: Considerations for dynamic analysis

Any of the direct time integration methods available in ADINA, except for the central difference method, can be used in direct time integration. Rayleigh damping can be specified in the structure.

Please remember that there is no damping within the fluid. Therefore the choice of initial conditions is extremely important, because poorly chosen initial conditions will cause transient responses that are not damped out.

2.11.18 Modeling: Considerations for frequency analysis and modal participation factor analysis

Frequency analysis (and all analyses that depend upon frequency data) cannot be performed if infinite interface elements are present in the analysis.

For ground motion modal participation factor calculations, it is recommended that each fluid region be completely surrounded by interface elements. This sets the modal participation factors for the ϕ rigid body modes to zero.

Figure 2.11-17 shows a one-dimensional example in which there is a ϕ rigid body mode and the ground motion modal participation factor is nonzero. The motion excited by the nonzero ground motion modal participation factor corresponds to a constant ground acceleration. Notice that this ground motion causes the unbounded expansion of the fluid region.

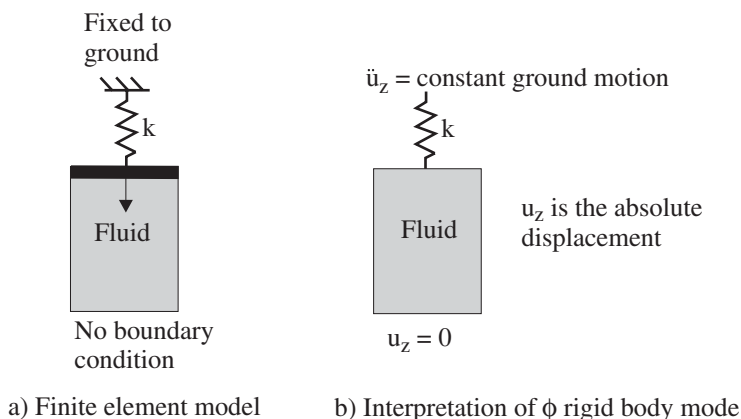


Figure 2.11-17: Problem in which the ground motion modal participation factor for the ϕ rigid body mode is nonzero

2.11.19 Modeling: Element output

Each fluid element outputs, at its integration points, the following information to the porthole file (the integration point numbering is the same as the numbering convention used for the solid elements): FE_PRESSURE, FLUID_REFERENCE_PRESSURE, ELEMENT_X-VELOCITY, ELEMENT_Y-VELOCITY, ELEMENT_Z-VELOCITY (ELEMENT_X-VELOCITY is applicable only for 3-D elements). FE_PRESSURE is the value of p and FLUID_REFERENCE_PRESSURE is the value of $\rho\Omega$ (see Equation (2.11-59)).

In the output of modal quantities, ELEMENT_X-VELOCITY, ELEMENT_Y-VELOCITY, ELEMENT_Z-VELOCITY should be interpreted as modal particle displacements, not velocities, in accordance with equation (2.11-42). However in all other types of analysis (static, direct time integration, modal superposition, response spectrum, harmonic and random), these quantities are velocities. The velocities are absolute velocities, even when ground motions are entered using mass-proportional loads.

Interface elements do not have any output.

2.12 Alignment elements

2.12.1 Overview

- The alignment element is a two-node element, as shown in Fig 2.12-1. The element is used to prescribe the relative translations, distances and/or rotations of the two nodes.
- Possible applications for the element include:
 - ▶ Bringing two bodies together
 - ▶ Robot arms and other parts in which the relative rotations are prescribed
 - ▶ Simulation of body-body contact
- There are three types of alignments
 - ▶ Translation alignments
 - ▶ Distance alignments
 - ▶ Rotation alignments

Any or all of these alignments can be used in the alignment element definition.

- Each alignment can be prescribed as a function of time. Thus the alignment element has some of the characteristics of a prescribed load or prescribed constraint boundary condition.

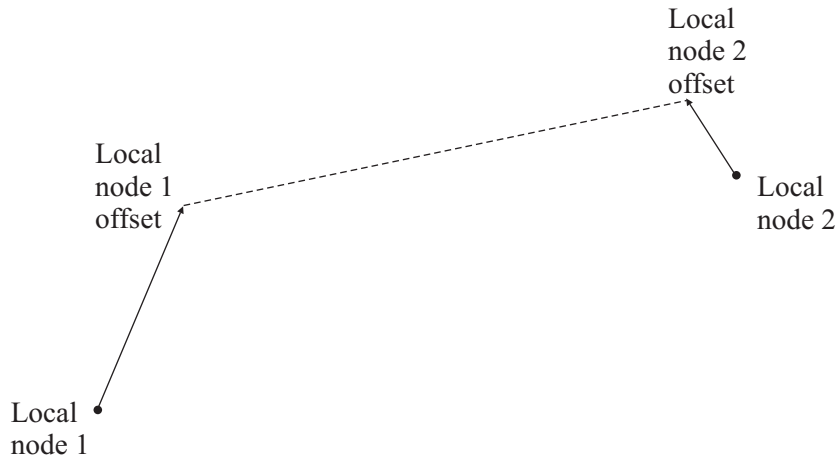


Figure 2.12-1: Alignment element

- In order to introduce the concepts used in the alignment element, we present some illustrative examples. Details are given later on.

Example 1) Translational alignment: As a simple example, consider an alignment element that aligns translations only. In Fig 2.12-2, there is a misalignment between the two nodes. This misalignment is measured by the vector connecting the two nodes. Also shown in this figure are equal and opposite forces generated by the alignment element. These forces are proportional to the misalignment, and these forces act to reduce the misalignment. The element provides stiffness in all three translational directions.

This element can be used to keep the two nodes nearly coincident. Notice that, although the nodes are nearly coincident, their relative rotations are not affected by the element, so the nodal rotations can be very different.

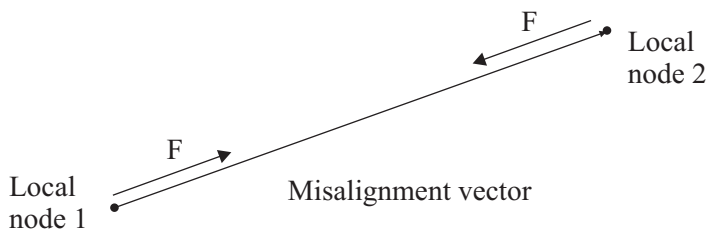


Figure 2.12-2: Example 1, translational alignment

Example 2) Translational alignment with prescribed translational alignment: This example considers the same alignment element as in Example 1, but now a prescribed alignment is included (Fig 2.12-3). The prescribed alignment is a vector measured from local node 1. The misalignment is the vector difference of the total alignment and the prescribed alignment. The forces generated by the alignment element are equal and opposite, and these forces act to reduce the misalignment. Notice that the forces do not in general act along the line connecting the two nodes, in particular a moment is generated by the alignment element force acting on local node 1.

The prescribed alignment is a function of time, entered as part of the definition of the translation alignment.

As a common special case, the prescribed alignment can be specified as a fraction of the initial alignment. This case is useful when the two nodes are initially misaligned, and the nodes are to be brought into alignment through several solution steps. In each solution step, the fraction is reduced, until the fraction is zero; then the prescribed alignment is zero.

In Fig 2.12-3(b), the prescribed alignment is equal to the initial misalignment. Therefore the misalignment is zero and no forces are generated by the alignment element. In Fig 2.12-3(c), the prescribed alignment is reduced, therefore the misalignment is non-zero and forces are generated. The forces will cause the nodes to approach each other.

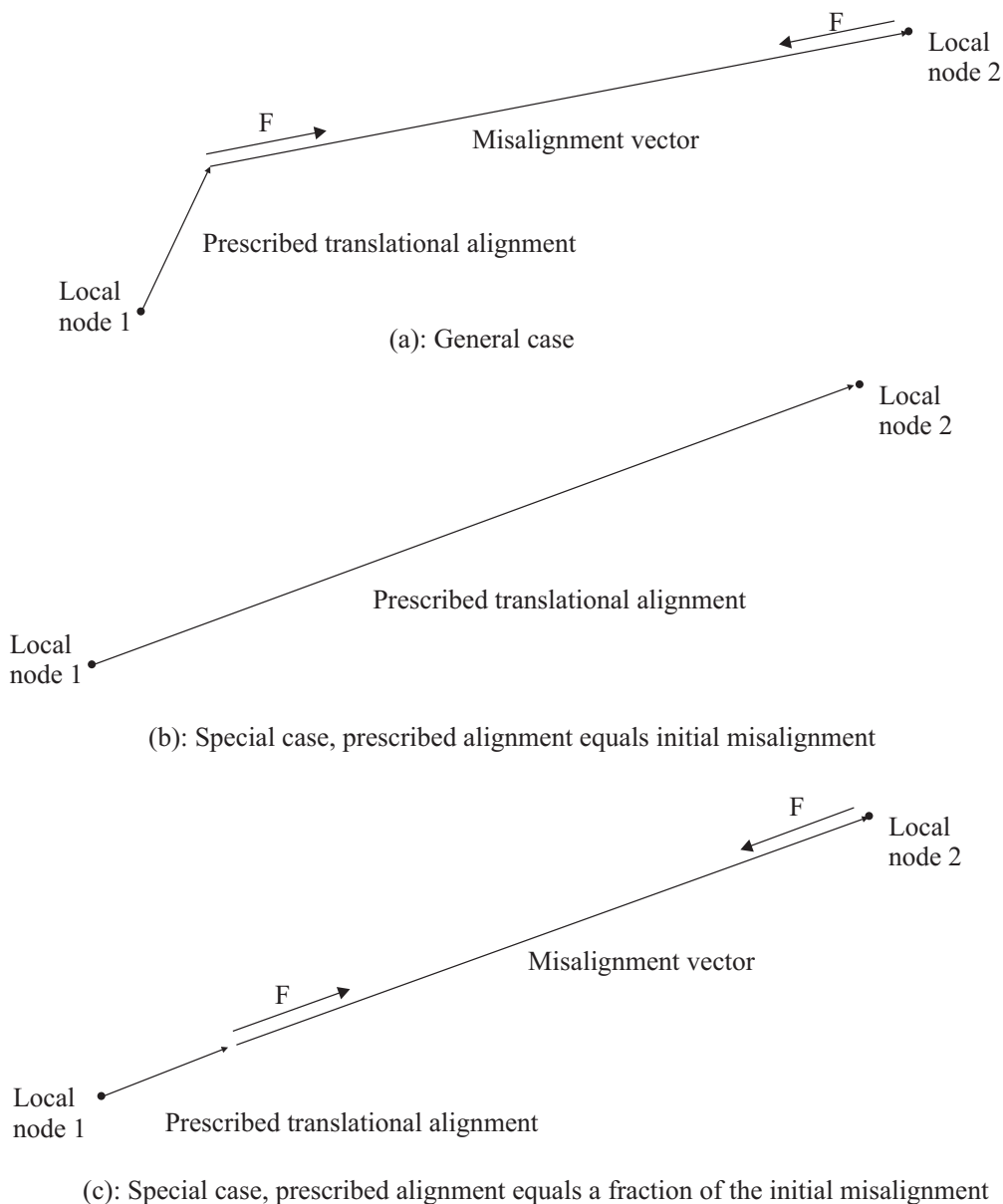


Figure 2.12-3: Example 2, translational alignment with prescribed alignment

Example 3) Translational alignment with prescribed translational alignment and nodal offsets: This example considers the same alignment element as in Example 2, but now nodal offsets are

included (Fig 2.12-4). Thus the prescribed alignment, total alignment and misalignments are measured from positions 1b and 2b. As in Example 2, the forces generated by the alignment element are equal and opposite, and these forces act to reduce the misalignment. Notice that the forces do not in general act along the line connecting the two nodes, and in this example moments act on both local nodes 1 and 2.

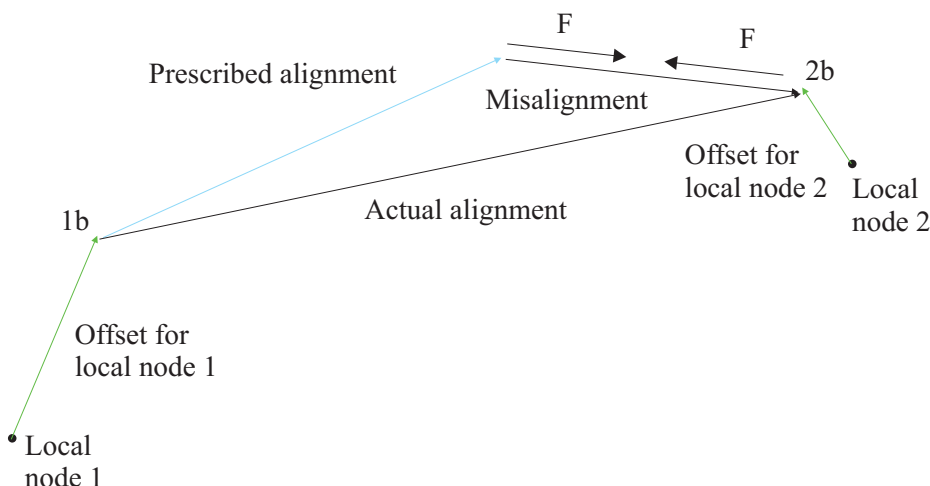


Figure 2.12-4: Example 3: Translational alignment with prescribed alignment and nodal offsets

Example 4) Distance alignment with prescribed distance alignment and nodal offsets: This example considers an alignment element that aligns distance, instead of aligning the translational components (Fig 2.12-5). Here the prescribed distance is a scalar quantity, measured from local node 1. The distance misalignment is the scalar difference of the total distance and the prescribed distance. The forces generated by the alignment element are equal and opposite, and these forces act to reduce the misalignment.

Of course, it is also possible to not use nodal offsets, then the distance is measured directly between the two nodes.

For the distance alignment, the force always acts along the line connecting points 1b and 2b. The element provides stiffness only in the direction of the line connecting points 1b and 2b.

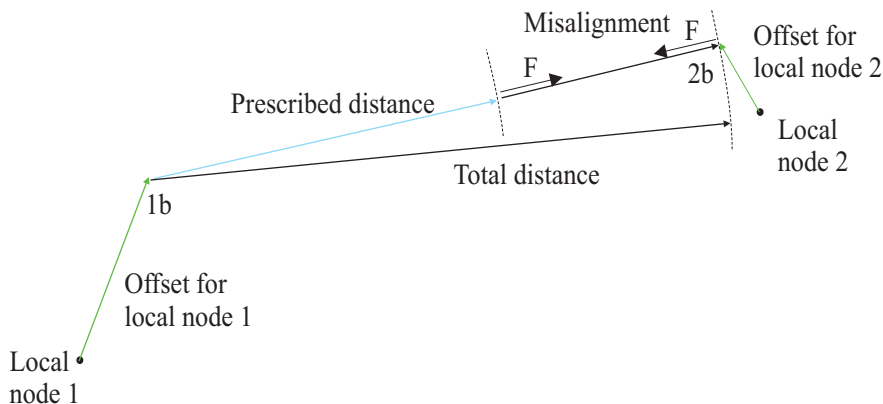


Figure 2.12-5: Example 4: Distance alignment with prescribed alignment and nodal offsets

Example 5) Rotation alignment: The discussion of rotational alignments is more difficult than the discussion of translational or distance alignments. In order to provide simple examples, we consider only planar rotations here. The general case of 3D rotations is discussed below.

Fig 2.12-6 shows an example of rotational alignments in the plane.

Here each local node is seen to have a local coordinate system, called a triad, attached to it. The directions of the triad axes for local node 1 are denoted $\mathbf{a1}^1$, $\mathbf{a2}^1$ and the directions of the triad axes for local node 2 are denoted $\mathbf{a1}^2$, $\mathbf{a2}^2$. The angle giving the orientation of the directions $\mathbf{a1}^2$, $\mathbf{a2}^2$ in terms of the directions $\mathbf{a1}^1$, $\mathbf{a2}^1$ is the rotational misalignment angle θ . This misalignment creates equal and opposite moments acting onto the two nodes.

As the nodes rotate, the triads also rigidly rotate.

The alignment element in this example can be used to keep the rotations of the two nodes nearly coincident. Notice that, although the rotations of the nodes are nearly coincident, their translations are not affected by the element, so the nodal positions need not be coincident.

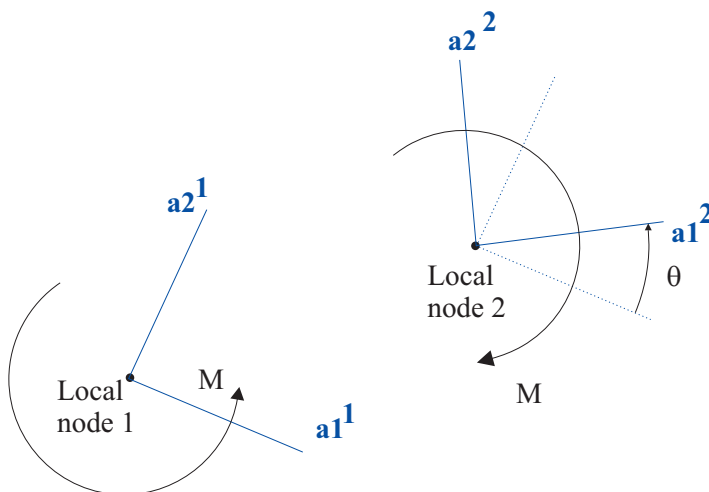


Figure 2.12-6: Example 5: Rotation alignment

Example 6) Rotation alignment with prescribed rotational alignment: Fig 2.12-7 shows another example of a rotational alignment element, here including a prescribed rotational alignment γ . Angle γ is used to construct another triad, with directions denoted $\mathbf{d1}^1, \mathbf{d1}^2$, relative to the directions $\mathbf{a1}^1, \mathbf{a2}^1$. The rotational misalignment angle θ is measured from directions $\mathbf{d1}^1, \mathbf{d1}^2$ to directions $\mathbf{a1}^2, \mathbf{a2}^2$. Again, the misalignment creates equal and opposite moments acting onto the two nodes.

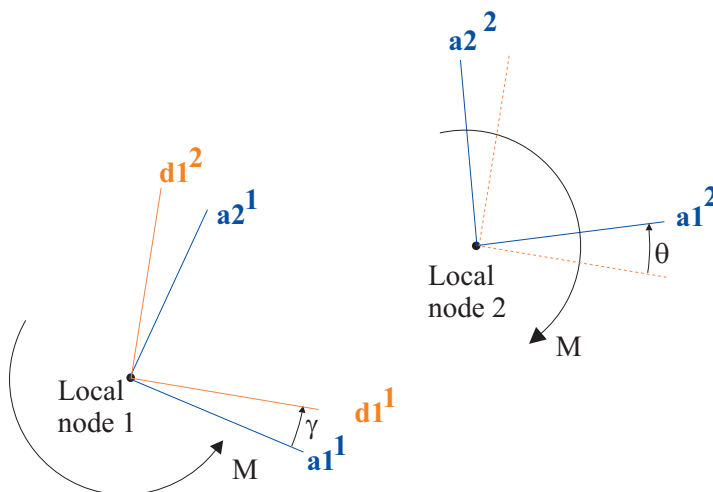


Figure 2.12-7: Example 6: Rotation alignment with prescribed rotational alignment

Example 7) Rotation alignment with prescribed rotational alignment and offset triads: Figure 2.12-8 shows another example of a rotational alignment element, here including both a prescribed rotational alignment and offset triads. The offset triad for node 1 is given by directions $\mathbf{b1}^1, \mathbf{b2}^1$ and the offset triad for node 2 is given by directions $\mathbf{b1}^2, \mathbf{b2}^2$. The prescribed rotational alignment γ is measured from directions $\mathbf{b1}^1, \mathbf{b2}^1$ and the rotational misalignment angle θ is measured from directions $\mathbf{d1}^1, \mathbf{d1}^2$ to directions $\mathbf{b1}^2, \mathbf{b2}^2$.

Although in the figure, the origins of the b triads is different than the origins of the a triads, it is certainly possible to have the origins of the b triads coincident with the origins of the a triads, and to have the b triad directions different than the a triad directions.

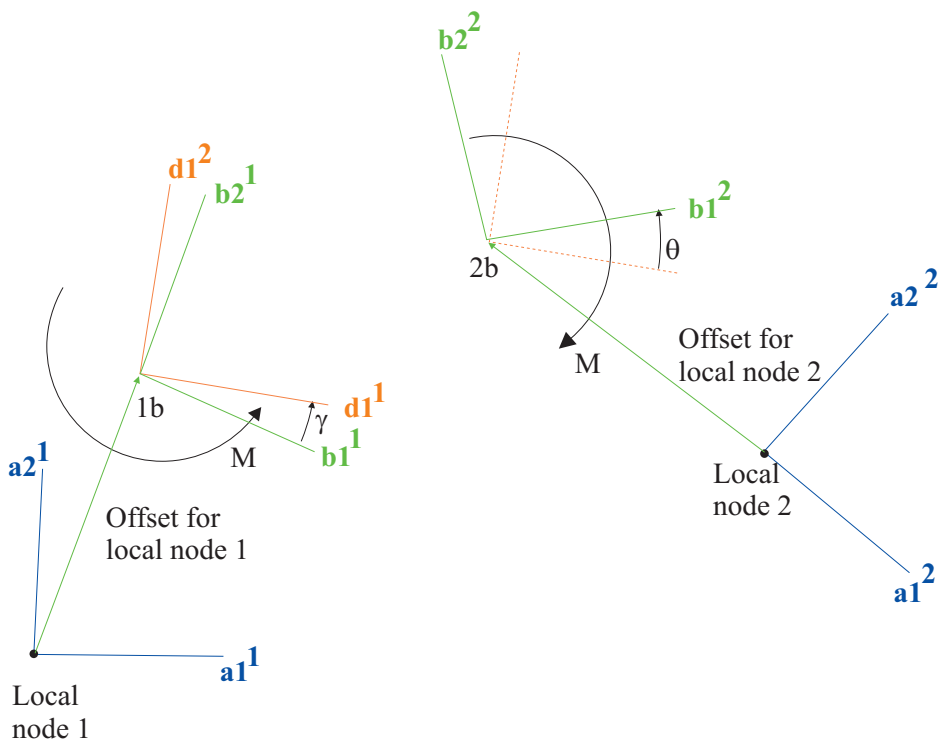


Figure 2.12-8: Example 7: Rotation alignment with prescribed rotational alignment and offset triads

- A single alignment element can align translations and/or distances and/or rotations. For example, a single element that aligns both translations and rotations can be used to keep two nodes nearly coincident, and also the relative rotations of the two nodes nearly equal.

2.12.2 Triads

In this section the nodal offsets and triads are discussed in more detail.

Figure 2.12-9 shows the triads at a node.

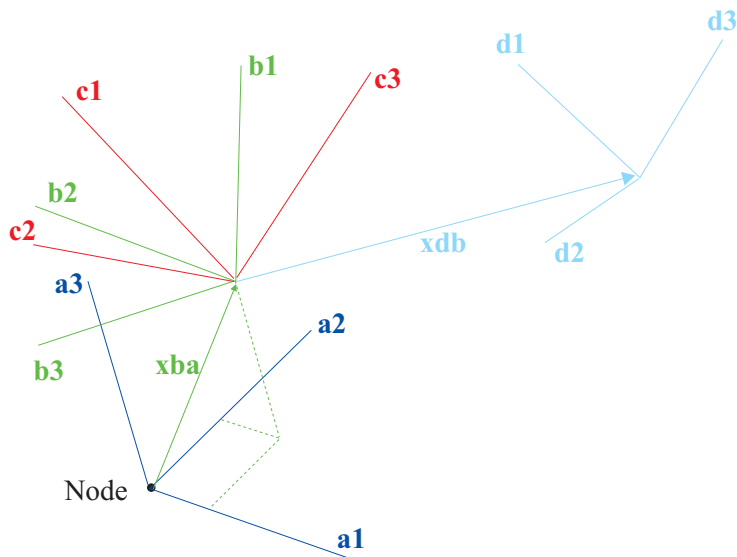


Figure 2.12-9: Triads at a node

A triad

Each node of the alignment element has a triad attached to the node. This triad is called the "a" triad, and the coordinate axes of the a triad are denoted **a1**, **a2**, **a3**. This triad is an orthonormal right-handed coordinate system, with origin at the node. As the node translates and rotates, the triad rigidly translates and rotates with the node.

B triad

Each node of the alignment element also has a triad which is in general offset from the a triad. This offset triad is called the "b" triad, and the coordinate axes of the b triad are denoted **b1**, **b2**, **b3**. The b triad is an orthonormal right-handed coordinate system, with the origin of the b triads given by vector **xba**.

The origin and coordinate axes of the b triad are defined with reference to the a triad. Thus the b triad is rigidly attached to the a triad. In particular, as the node rotates, vectors **xba**, **b1**, **b2**, **b3** also rotate.

C triad

Local node 1 of the alignment element also has a triad that provides a coordinate system in which the prescribed alignments, total alignments and misalignments are measured. This triad is called the "c" triad and the coordinate axes of the c triad are denoted **c1**, **c2**, **c3**. The c triad is an orthonormal right-handed coordinate system, with origin at the origin of the b triad. The coordinate axes of the c triad are defined with reference to the b triad. Thus the c triad is rigidly attached to both the b triad and the a triad.

Thus, for example, the components of the prescribed translational misalignments are specified in the c triad coordinate system.

D triad

Local node 1 of the alignment element also has a triad that represents the prescribed translation or rotation. This triad is called the "d" triad and the coordinate axes of the d triad are denoted **d1**, **d2**, **d3**. The d triad is an orthonormal right-handed coordinate system, with origin offset from the origin of the b triad.

The prescribed translational alignment is used to determine the origin of the d triad relative to the b triad of local node 1 (in other words, the vector **xdb**), and the prescribed rotational alignment is used to determine the orientation of the d triad relative to the orientation of the b triad of local node 1. This will be discussed in more detail later.

2.12.3 Notations

The notations used in the discussion of the theory are based on the triad names (a, b, c, d). Although the notations are cumbersome, the notations clearly show the meanings of the quantities described.

Vectors

xba : vector from a triad origin to b triad origin

xdb : vector from b triad origin to d triad origin

When it is necessary to denote a particular local node, a superscript is used:

$\mathbf{x}^1\mathbf{a}^1$: vector from a triad origin of local node 1 to b triad origin of local node 1

$\mathbf{x}^2\mathbf{a}^2$: vector from a triad origin of local node 2 to b triad origin of local node 2

$\mathbf{x}^1\mathbf{b}^1$: vector from b triad origin of local node 1 to d triad origin of local node 1

$\mathbf{x}^2\mathbf{d}^1$: vector from d triad origin of local node 1 to b triad origin of local node 2.

Obviously the laws of vector addition hold, for example

$$\mathbf{x}^1\mathbf{a}^1 = \mathbf{x}^1\mathbf{b}^1 + \mathbf{x}^2\mathbf{a}^1.$$

Since the above quantities are vectors, they can be expressed using components in any convenient coordinate system. In order to denote which coordinate system the components are expressed in, we use a letter in the subscripts, for example $(x^1b_{c1}^1, x^1b_{c2}^1, x^1b_{c3}^1)$ are the components of vector $\mathbf{x}^1\mathbf{b}^1$, expressed in the c triad coordinate system.

Rotation matrices

For rotations, we need to express the directions of one triad in terms of the directions of another triad. For example

$$\mathbf{V}^2\mathbf{B}^1 = \begin{bmatrix} b1_{b1}^2 & b1_{b2}^2 & b1_{b3}^2 \\ b2_{b1}^2 & b2_{b2}^2 & b2_{b3}^2 \\ b3_{b1}^2 & b3_{b2}^2 & b3_{b3}^2 \end{bmatrix}$$

The first row of the matrix gives the direction $\mathbf{b1}$ for local node 2, the second row gives the direction $\mathbf{b2}$ for local node 2, and the third row gives the direction $\mathbf{b3}$ for local node 2. Each of these direction vectors is expressed using the coordinate system of the b triad of local node 1.

Clearly if the directions of the b triad axes are the same for both

local nodes, then there is no relative rotation between the b triads of local nodes 1 and 2, and this matrix reduces to the identity matrix.

We also use

$$\mathbf{VB}^2\mathbf{D}^1 = \begin{bmatrix} b1_{d1^1}^2 & b1_{d2^1}^2 & b1_{d3^1}^2 \\ b2_{d1^1}^2 & b2_{d2^1}^2 & b2_{d3^1}^2 \\ b3_{d1^1}^2 & b3_{d2^1}^2 & b3_{d3^1}^2 \end{bmatrix} : \text{ the b triad directions for local}$$

node 2 in terms of the d triad directions for local node 1

$$\mathbf{VD}^1\mathbf{B}^1 = \begin{bmatrix} d1_{b1^1}^1 & d1_{b2^1}^1 & d1_{b3^1}^1 \\ d2_{b1^1}^1 & d2_{b2^1}^1 & d2_{b3^1}^1 \\ d3_{b1^1}^1 & d3_{b2^1}^1 & d3_{b3^1}^1 \end{bmatrix} : \text{ the d triad directions for local}$$

node 1 in terms of the b triad directions for local node 1

The rules of matrix multiplication apply, for example

$\mathbf{VB}^2\mathbf{B}^1 = \mathbf{VB}^2\mathbf{D}^1 \cdot \mathbf{VD}^1\mathbf{B}^1$. Also the inverse of the matrix is the same as the transpose, for example $(\mathbf{VD}^1\mathbf{B}^1)^{-1} = (\mathbf{VD}^1\mathbf{B}^1)^T$.

2.12.4 Translation alignment theory

Figure 2.12-10 shows an alignment element that uses a translation alignment.

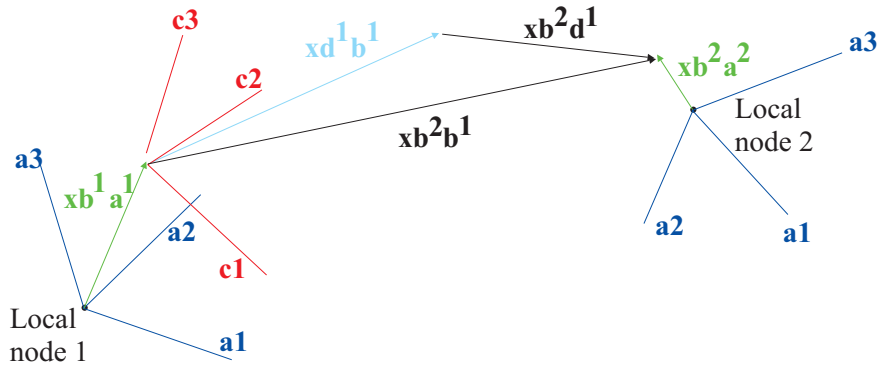


Figure 2.12-10: Translation vectors used in translation alignment

Translation alignment calculation

When the translation alignment is used, the vector between the origin of the b triad of local node 1 and the origin of the b triad of local node 2 is calculated ($\mathbf{xb}^2 \mathbf{b}^1$).

Notice that the directions of the b triad coordinate axes are not used in the translation calculation.

Prescribed translation

The user enters a table giving the "prescribed translation" for various times. This prescribed translation can be specified using the following options:

Aligned: The prescribed translation is set to zero.

Components: The prescribed translation is directly specified as the vector $\mathbf{xd}^1 \mathbf{b}^1$, given in terms of components in the c triad coordinate system ($xd^1 b_{c1}^1, xd^1 b_{c2}^1, xd^1 b_{c3}^1$).

Tbfactor: The prescribed translation is specified relative to the translations at the time of the birth of the alignment. The vector $\mathbf{xb}^2 \mathbf{b}^1$ is evaluated at the time of birth of the alignment, and this vector is used to construct a prescribed translation with the same

direction, but with a scaled value of the magnitude, that is

$$\mathbf{x}\mathbf{d}^1\mathbf{b}^1 = \text{factor} \times \mathbf{x}\mathbf{b}^2\mathbf{b}^1 \Big|_{\text{birth}}.$$

Same: The prescribed translation is the same as in the previous row of the align-translation table. (This option cannot be used in the first row of the align-translation table.)

Notice that the directions of the d triad coordinate system are not used in this calculation.

The ALIGN-TRANSLATION command is used to enter this table.

Translation misalignment

The difference between the actual translation and the prescribed translation is calculated using $\mathbf{x}\mathbf{b}^2\mathbf{d}^1 = \mathbf{x}\mathbf{b}^2\mathbf{b}^1 - \mathbf{x}\mathbf{d}^1\mathbf{b}^1$. This difference is termed the translation misalignment. Then vector $\mathbf{x}\mathbf{b}^2\mathbf{d}^1$ is expressed using its components in the c triad coordinate system: $(xb^2d_{c1}^1, xb^2d_{c2}^1, xb^2d_{c3}^1)$

Forces

The alignment element calculates a force from the translation misalignment in each coordinate direction:

$F_{c1} = k_{c1} \cdot xb^2d_{c1}^1, F_{c2} = k_{c2} \cdot xb^2d_{c2}^1, F_{c3} = k_{c3} \cdot xb^2d_{c3}^1$. Notice that the stiffness can be different in each coordinate direction. Hence, for example, the user might specify that the stiffness be zero for any misalignment in the c1 coordinate direction, but nonzero for misalignments in the other coordinate directions. Figure 2.12-11 shows one possibility, drawn in two dimensions for simplicity.

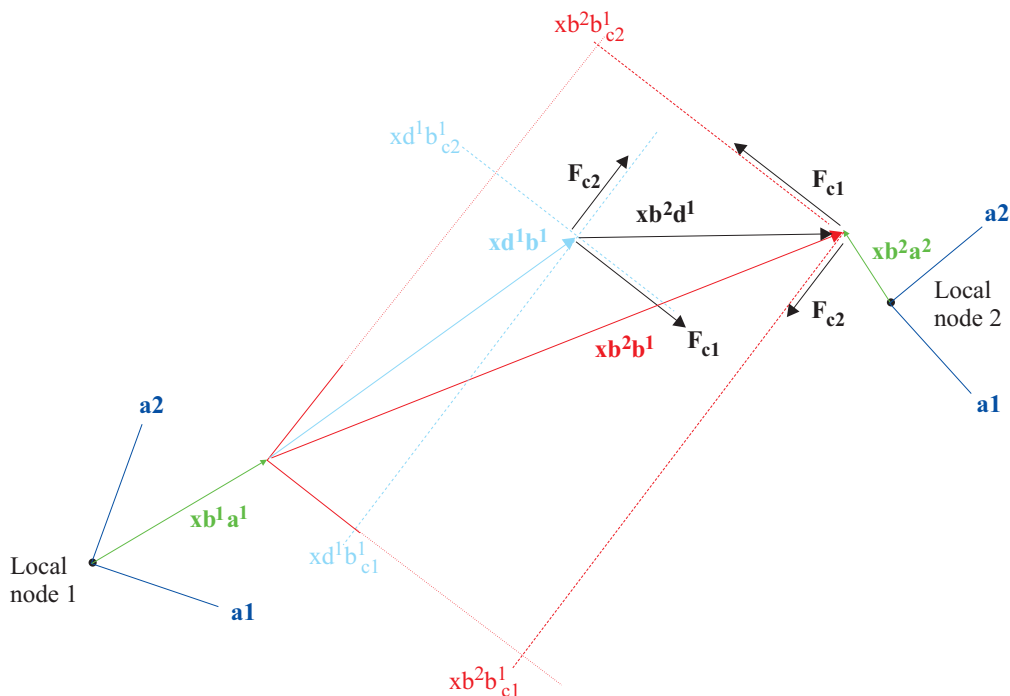


Figure 2.12-11: Prescribed translations and forces for translation alignment, two-dimensional case

The stiffness coefficients are entered in the ALIGN-TRANSLATION command.

All forces act to reduce the translation misalignments.

The principle of virtual work is used to obtain the corresponding forces and moments acting at the nodes.

2.12.5 Distance alignment theory

Fig. 2.11-12 shows an alignment element that uses a distance alignment.

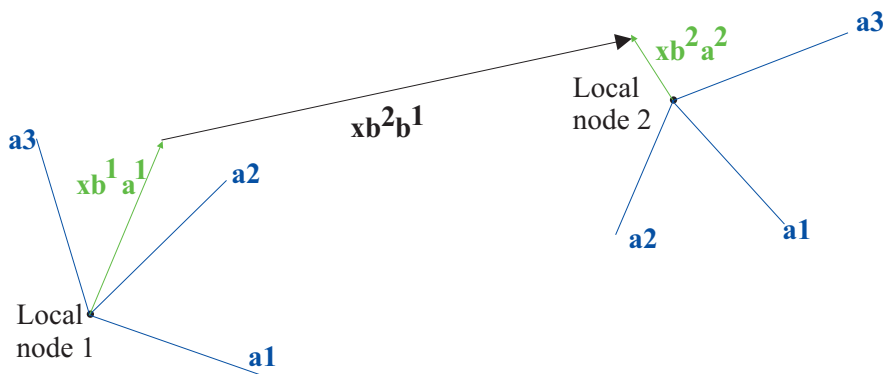


Figure 2-12.12: Distance vector used in distance alignment

Distance calculation

When the distance alignment is used, the distance D is measured using the length of vector xb^2b^1 . Vector xb^2b^1 connects the origin of the b triad of node 2 with the origin of the b triad of node 1. Notice that the directions of the b triad axes are not used.

Also notice that the distance changes if the nodes rotate without translation, since the offset vectors xb^1a^1 , xb^2a^2 also rotate as the nodes rotate.

Prescribed distance

The user enters a table giving the "prescribed distance" for various times. The prescribed distance can be specified using the following options:

Aligned: The prescribed distance is set to zero.

Distance: The prescribed distance is directly specified as the number D^p .

Tbfactor: The prescribed distance is specified relative to the distance at the time of the birth of the alignment. D is evaluated at the time of birth of the alignment, and this vector is used to construct a prescribed distance with a scaled value of the

magnitude, that is $D^P = \text{factor} \times D|_{\text{birth}}$.

Same: The prescribed distance is the same as in the previous row of the align-distance table. (This option cannot be used in the first row of the align-distance table.)

The ALIGN-DISTANCE command is used to enter this table.

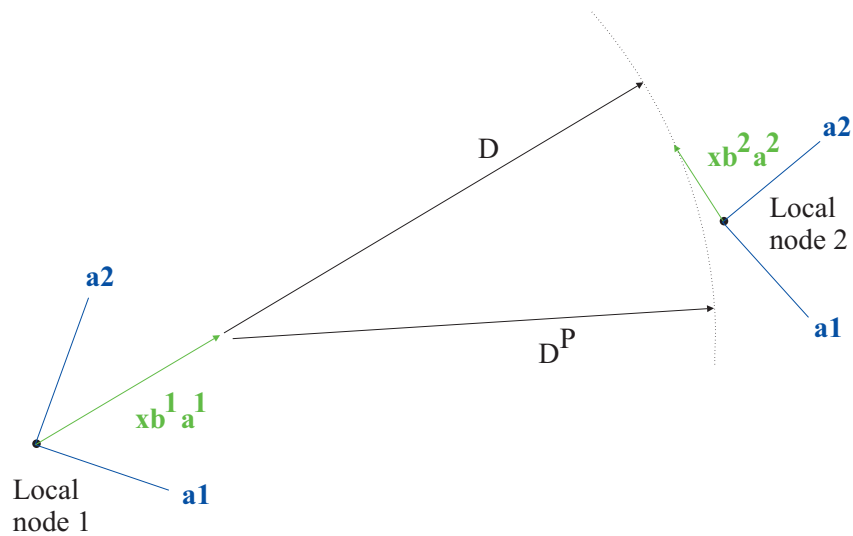
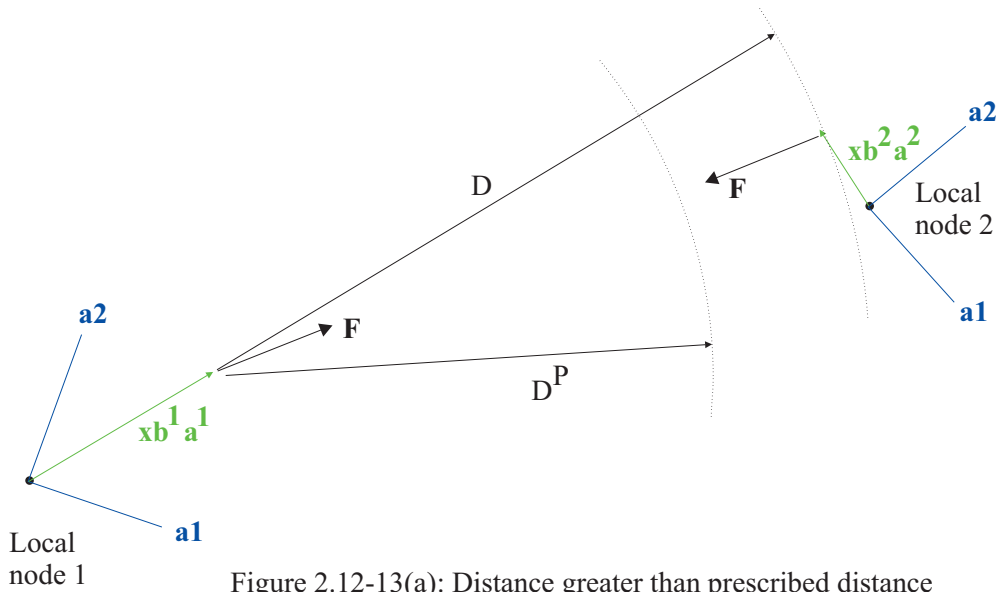
Distance misalignment

$D - D^P$ is termed the distance misalignment.

Forces

The alignment element calculates a force from the difference between the distance and the prescribed distance $F_d = k_d (D - D^P)$.

This force acts along the line connecting the origins of the b triads, and acts to reduce the difference between the distance and the prescribed distance. Fig. 2.12-13 shows some possibilities, drawn in two dimensions for simplicity.



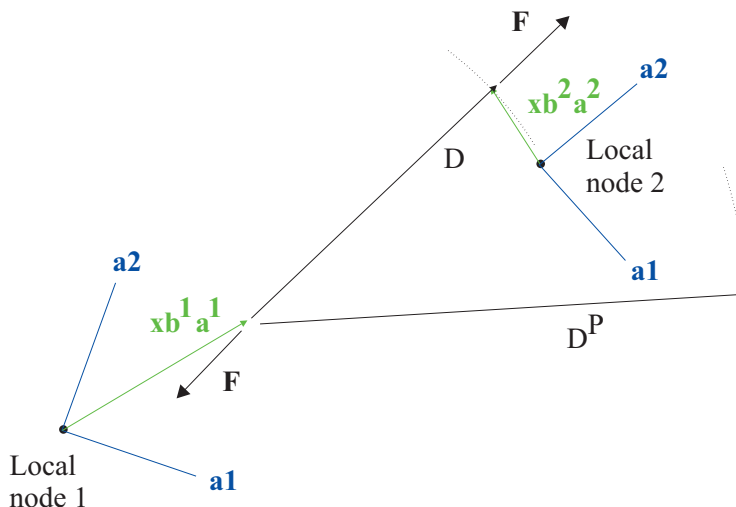


Figure 2.12-13(c): Distance less than prescribed distance

Figure 2.12-13: (continued)

The stiffness coefficient is entered in the ALIGN-DISTANCE command.

The principle of virtual work is used to obtain the corresponding forces and moments acting at the nodes.

2.12.6 Rotation alignment theory

Fig. 2.12-14 shows an alignment element that uses a rotation alignment. Only the directions of the triads are used, and not their origins, as shown in Fig. 2.12-14(b).

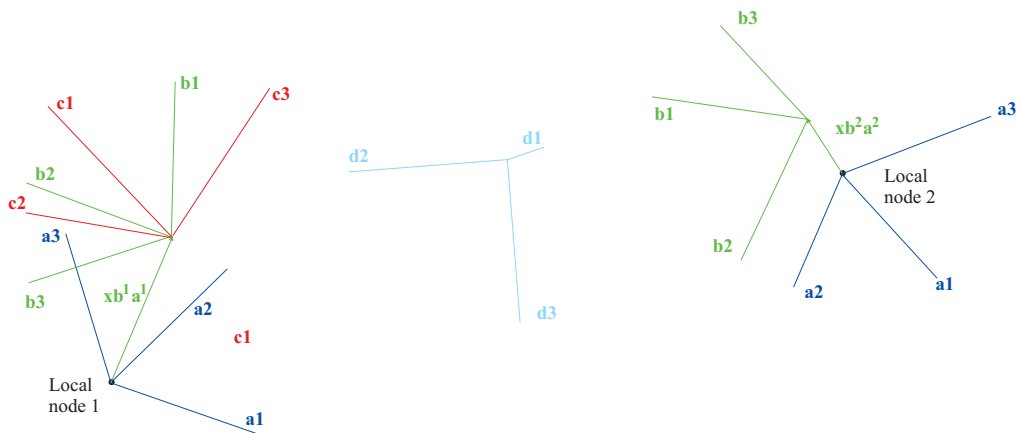


Figure 2.12-14(a): Rotation alignment

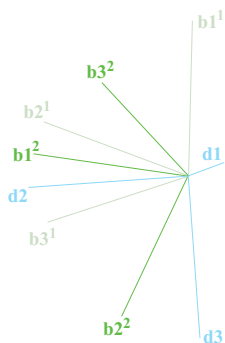


Figure 2.12-14(b): Relative orientation of b and d triads, corresponding to (a)

Rotation calculation

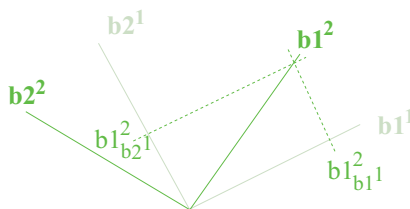
When the rotation alignment is used, the relative rotation between the b triad of local node 2 and the b triad of local node 1 is

$$\text{calculated: } \mathbf{VB}^2\mathbf{B}^1 = \begin{bmatrix} b1_{b1^1}^2 & b1_{b2^1}^2 & b1_{b3^1}^2 \\ b2_{b1^1}^2 & b2_{b2^1}^2 & b2_{b3^1}^2 \\ b3_{b1^1}^2 & b3_{b2^1}^2 & b3_{b3^1}^2 \end{bmatrix}.$$

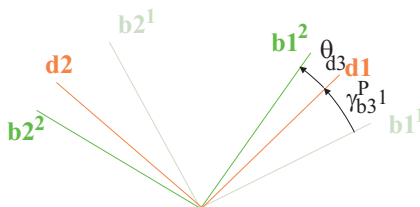
It is easier to visualize the relative rotation for the special case in which the b3 axes of the two triads have the same direction. This case is shown in Fig 2.12-15(a). In this special case, the matrix has

the form

$$\mathbf{VB}^2\mathbf{B}^1 = \begin{bmatrix} b1^2_{b1^1} & b1^2_{b2^1} & 0 \\ b2^2_{b1^1} & b2^2_{b2^1} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



(a): Relative orientation of b triads



(b): Prescribed rotations and rotation misalignment

Figure 2.12-15: Rotations in the plane, b3 axes have the same direction

Prescribed rotation

The user specifies the "prescribed rotation" for various times. Internally, the prescribed rotation is expressed as a coordinate triad **d1**, **d2**, **d3**, defined relative to the b triad of node 1. In matrix form, the prescribed rotation can be written as

$$\mathbf{VD}^1\mathbf{B}^1 = \begin{bmatrix} d1_{b1^1}^1 & d1_{b2^1}^1 & d1_{b3^1}^1 \\ d2_{b1^1}^1 & d2_{b2^1}^1 & d2_{b3^1}^1 \\ d3_{b1^1}^1 & d3_{b2^1}^1 & d3_{b3^1}^1 \end{bmatrix}$$

Again, this situation is easier to visualize for the special case in which the b3 axes of the two triads, and also the d3 axis, have the same directions. This case is shown in Fig 2.12-15(b).

The prescribed rotation can be specified using the following options:

Aligned: $\mathbf{VD}^1\mathbf{B}^1 = \mathbf{I}$

Angle: The user enters an angle γ^P and axis of rotation $(n_{c1^1}^P, n_{c2^1}^P, n_{c3^1}^P)$. The program normalizes the axis of rotation, then the program calculates the axis of rotation in the b triad system of node 1 $(n_{b1^1}^P, n_{b2^1}^P, n_{b3^1}^P)$. Next the program calculates $(\gamma_{b1^1}^P, \gamma_{b2^1}^P, \gamma_{b3^1}^P) = \gamma^P (n_{b1^1}^P, n_{b2^1}^P, n_{b3^1}^P)$ and then calculates the prescribed rotation matrix using the formula

$$\mathbf{VD}^1\mathbf{B}^1 = \begin{bmatrix} d1_{b1^1} & d1_{b2^1} & d1_{b3^1} \\ d2_{b1^1} & d2_{b2^1} & d2_{b3^1} \\ d3_{b1^1} & d3_{b2^1} & d3_{b3^1} \end{bmatrix} = \mathbf{I} + \frac{\sin \gamma^P}{\gamma^P} \mathbf{S} + \frac{1}{2} \left(\frac{\sin \frac{\gamma^P}{2}}{\frac{\gamma^P}{2}} \right) \mathbf{S}^2,$$

$$\text{where } \mathbf{S} = \begin{bmatrix} 0 & -\gamma_{b3^1}^P & \gamma_{b2^1}^P \\ \gamma_{b3^1}^P & 0 & -\gamma_{b1^1}^P \\ -\gamma_{b2^1}^P & \gamma_{b1^1}^P & 0 \end{bmatrix}$$

Notice that the components of the axis of rotation are specified in the coordinate system of the c triad of local node 1. Also notice that angle γ^P and axis of rotation $(n_{c1^1}^P, n_{c2^1}^P, n_{c3^1}^P)$ represents the total angle and axis corresponding to the prescribed rotation matrix. If the prescribed rotation is expressed as a series of incremental rotations, the incremental rotations must be combined, then converted back into the form

$$\gamma^P, \left(n_{c1^1}^P, n_{c2^1}^P, n_{c3^1}^P \right).$$

Relative to rotations at time of birth: The matrix

$$\mathbf{VB}^2\mathbf{B}^1 = \begin{bmatrix} b1_{b1^1}^2 & b1_{b2^1}^2 & b1_{b3^1}^2 \\ b2_{b1^1}^2 & b2_{b2^1}^2 & b2_{b3^1}^2 \\ b3_{b1^1}^2 & b3_{b2^1}^2 & b3_{b3^1}^2 \end{bmatrix} \text{ is evaluated at the time of birth}$$

of the alignment, and this matrix is used to construct a prescribed rotation $\mathbf{VD}^1\mathbf{B}^1$ with the same axis, but with a scaled value of the angle.

Incremental: The matrix $\mathbf{VD}^1\mathbf{B}^1$ is computed by incremental update of the matrix $\mathbf{VD}^1\mathbf{B}^1$ for the previous row of the table. Specifically

$$\mathbf{VD}^1\mathbf{B}^1 = \left[\mathbf{I} + \frac{\sin \gamma}{\gamma} \mathbf{S} + \frac{1}{2} \left(\frac{\sin \frac{\gamma}{2}}{\frac{\gamma}{2}} \right) \mathbf{S}^2 \right] \mathbf{VD}^1\mathbf{B}^1 \Big|_{\text{previous row}}$$

where \mathbf{S} , $\gamma^P \left(n_{c1^1}^P, n_{c2^1}^P, n_{c3^1}^P \right)$ are defined above. If this option is used for the first row of the table, then the effect is the same as if option=Angle.

Same: The prescribed rotation is the same as in the previous row of the align-rotation table. (This option cannot be used in the first row of the align-rotation table.)

The ALIGN-ROTATION command is used to enter this table.

When angles are specified, the unit is either radians or degrees, depending on the value of ALIGN-ROTATION UNITANGLE (the default is radians).

Rotation misalignment

The rotation misalignment can be calculated as the product of the actual rotation and the inverse of the prescribed rotation:

$$\mathbf{VB}^2\mathbf{D}^1 = \mathbf{VB}^2\mathbf{B}^1 \cdot (\mathbf{VD}^1\mathbf{B}^1)^T$$

The rotation misalignment matrix $\mathbf{VB}^2\mathbf{D}^1$ is then represented as a rotation misalignment angle and axis: angle θ^m and axis of rotation $(n_{d1}^m, n_{d2}^m, n_{d3}^m)$, where the superscript m stands for "misalignment".

The axis of rotation is then transformed from the d1 triad system to the c1 triad system to obtain $(n_{c1}^m, n_{c2}^m, n_{c3}^m)$.

Moments

The rotation misalignment angle and axis can also be written in component form:

$$(\theta_{c1}, \theta_{c2}, \theta_{c3}) = \theta^m (n_{c1}^m, n_{c2}^m, n_{c3}^m)$$

The moments are then calculated using the rotation misalignment vector:

$$M_{c1} = k_{c1} \cdot \theta_{c1}, M_{c2} = k_{c2} \cdot \theta_{c2}, M_{c3} = k_{c3} \cdot \theta_{c3}.$$

Notice that the rotational stiffness can be different in each coordinate direction. Hence, for example, the user might specify that the stiffness be zero for any rotational misalignment in the c1 coordinate direction, but nonzero for misalignments in the other coordinate directions.

The unit of k_{ci} is always moment/radian.

It is also allowed to enter a sawtooth moment / angle relationship for any of the coordinate directions. The sawtooth moment / angle relationship is shown in Fig 2-12.16.

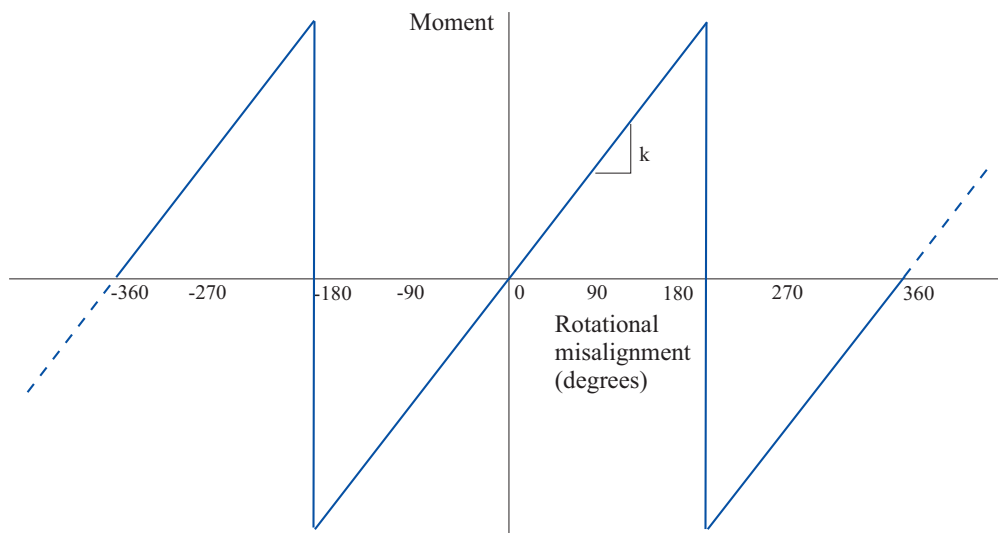


Figure 2.12-16: Sawtooth moment / angle relationship

Using this option, the moment is zero for misalignments of 0 degrees, 180 degrees, 360 degrees, etc.

The stiffness coefficients, and choices of constant or sawtooth stiffnesses, are entered in the ALIGN-ROTATION command.

All moments act to reduce the rotation misalignments.

2.12.7 Triadsets and node-triadset pairs

Triadsets

The combination of the a triad, b triad and c triad is called a triadset. The TRIADSET command is used to define the triads of the triad set.

The triadset consists of three parts:

Specification of the a triad: The choices are:

ELEMENTS: The a triad corresponds to the local axes of the element to which the node is attached. Currently, only

Hermitian beam elements contribute local axes to the a triad. Thus, for example, if the node is connected to a single Hermitian beam element, the a1 direction is the r direction of the beam element, the a2 direction is the s direction of the beam element, etc.

If the node is not attached to any element with local axes, the a triad directions are the global coordinate directions.

If the node is connected to two or more elements with local axes, then the program averages the local axes to obtain the a triad axes.

VECTORS: Components (a1x, a1y, a1z) and (a2x, a2y, a2z) are input. Then the a triad directions are computed using

a1 = (a1x, a1y, a1z), normalized to unit length

a3 = **a1** × (a2x, a2y, a2z), normalized to unit length

a2 = **a3** × **a1**

Specification of the b triad: The choices are

ATRIAD: The b triad coincides with the a triad.

VECTORS: The b triad origin is offset from the a triad origin using components (b0a1, b0a2, b0a3). Using the notation of 2.12.3, $(b0a1, b0a2, b0a3) = (xb^1a_{a1}^1, xb^1a_{a2}^1, xb^1a_{a3}^1)$.

The b triad directions are computed as follows: Components (b1a1, b1a2, b1a3) and (b2a1, b2a2, b2a3) are input. Then the b triad directions are computed using

b1 = (b1a1, b1a2, b1a3), normalized to unit length

b3 = **b1** × (b2a1, b2a2, b2a3), normalized to unit length

b2 = **b3** × **b1**

Specification of the c triad: The choices are:

BTRIAD: The c triad coincides with the b triad.

VECTORS: The c triad directions are computed as follows: Components (c1b1, c1b2, c1b3) and (c2b1, c2b2, c2b3) are

input. Then the c triad directions are computed using

$\mathbf{c1} = (\mathbf{c1b1}, \mathbf{c1b2}, \mathbf{c1b3})$, normalized to unit length

$\mathbf{c3} = \mathbf{c1} \times (\mathbf{c2b1}, \mathbf{c2b2}, \mathbf{c2b3})$, normalized to unit length

$\mathbf{c2} = \mathbf{c3} \times \mathbf{c1}$

Node-triadset pair

A triadset is assigned to a node using the SET-TRIADSET NODES command. There can be more than one triadset assigned to a node. Thus, for example, if a node is attached to more than one alignment element, one alignment element can use one triadset and the other alignment element can use a different triadset.

It is also allowed for no triadset to be assigned to a node. Then, if the node is attached to an alignment element, the program calculates a default triadset. The calculation of the default is discussed below.

2.12.8 Align-translation specification

The ALIGN-TRANSLATION command is used to define the characteristics of the translation alignment of an alignment element. These characteristics include the translational stiffnesses and the prescribed translational alignments.

The prescribed translational alignments are input in the form of a table. Each row of the table gives the prescribed translational alignment at a time. This time need not be a solution time.

For example, consider the command

```
ALIGN-TRANSLATION 1
ENTRIES TIME OPTION  XCP  YCP  ZCP  TBFACTOR
CLEAR
      3.0 TBFACTOR    0.0   0.0   0.0   1.0
      4.0 COMPONENTS 1.0   2.0   3.0   0.0
      5.0 ALIGNED
DATAEND
```

The lowest time in the table is time 3.0. For solution times before time 3.0, the translational alignment is inactive (not born yet).

For the first solution time at or after time 3.0, the translational alignment becomes active. The total alignment for the preceding solution time is measured and is used to construct $\mathbf{x}\mathbf{b}^2\mathbf{b}^1\big|_{birth}$. Since $OPTION=TBFACTOR$ and $TBFACTOR=1.0$, the prescribed translational alignment for time 3.0 is set to $\mathbf{x}\mathbf{d}^1\mathbf{b}^1 = \mathbf{x}\mathbf{b}^2\mathbf{b}^1\big|_{birth}$.

For solution time 4.0, the prescribed translational alignment is $(\mathbf{x}\mathbf{d}^1\mathbf{b}_{c1}^1, \mathbf{x}\mathbf{d}^1\mathbf{b}_{c2}^1, \mathbf{x}\mathbf{d}^1\mathbf{b}_{c3}^1) = (1.0, 2.0, 3.0)$.

For solution time 5.0, the prescribed translational alignment is $(\mathbf{x}\mathbf{d}^1\mathbf{b}_{c1}^1, \mathbf{x}\mathbf{d}^1\mathbf{b}_{c2}^1, \mathbf{x}\mathbf{d}^1\mathbf{b}_{c3}^1) = (0.0, 0.0, 0.0)$.

For solution times after time 5.0, the prescribed translational alignment is the same as the prescribed translational alignment at time 5.0.

For solution times between times given in the align-translation table, linear interpolation is used to determine the prescribed translational alignment.

2.12.9 Align-distance specification

The ALIGN-DISTANCE command is used to define the characteristics of the distance alignment of an alignment element. These characteristics include the distance stiffness and the prescribed distance alignments.

The prescribed distance alignments are input in the form of a table, analogous to the table used in the align-translation definition. The options are given in Section 2.12.5.

2.12.10 Align-rotation specification

The ALIGN-ROTATION command is used to define the characteristics of the rotation alignment of an alignment element. These characteristics include the rotational stiffnesses and the

prescribed rotational alignments.

The prescribed rotational alignments are input in the form of a table, analogous to the table used in the align-translation definition. The options are given in Section 2.12.6.

2.12.11 Element group specification

The EGROUP ALIGNMENT command is used to define the characteristics of an alignment element group. The default values of the align-translation, align-distance and align-rotation definitions can be entered in the EGROUP ALIGNMENT command.

2.12.12 Element specification

The nodes of the alignment element are specified using the ENODES command, just as for other element types. Notice that local node 1 is different than local node 2, because local node 1 is used to measure prescribed alignments. Thus two elements differing only in the order of the local nodes will in general give different results.

Data for the alignment element is specified using the EDATA command, just as for other element types. Note that it is allowed to enter the align-translation, align-distance and align-rotation definitions separately for each element. Therefore one element might align translations only and another element in the same group might align rotations only.

The data for the alignment element also includes the triadset to be used for each local node. This data is entered using parameters TRIADST1 and TRIADST2.

The assignment of a triadset to a local node is now discussed in detail for local node 1 (local node 2 is similar):

Case 1: The global node is not in a node-triadset pair:

The a triad is determined as follows:

Case 1a: Global node is attached to at least one element with local axes. Then the program averages the triads of all attached elements with local axes to obtain the a triad at the node.

Case 1b: Global node is not attached to any element with local axes. Then the program sets the a triad to the global coordinate directions.

The b triad is coincident with the a triad (with no offset).

The c triad is coincident with the b triad.

The TRIADST1 parameter must be set to zero.

Case 2: The global node is in a single node-triadset pair

The TRIADST1 parameter can either be set to zero, or to the triadset for the global node. The triadset is obtained from the node-triadset pair.

Case 3: The global node is in more than one node-triadset pair

The TRIADST1 parameter must be nonzero. This parameter is used to choose the triadset for the node. The triadset specified by parameter TRIADST1 must be in a node-triadset pair with the global node corresponding to local node 1.

2.12.13 Input examples

Example 1: In this example, nodes 1 and 2 are aligned using a distance alignment only. The nodes are to be aligned at time 1.0, with the two nodes coinciding.

```
TIMESTEP
CLEAR
1 1.0
DATAEND
*
ALIGN-DISTANCE 1
ENTRIES TIME OPTION
CLEAR
1.0 ALIGNED
DATAEND
*
```

```
EGROUP ALIGNMENT 1 ALIGN-DISTANCE=1
EDATA
ENTRIES N1 N2
CLEAR
1 2
DATAEND
```

Example 2: This example is similar to example 1. However, the alignment starts at time 1.0 with a prescribed alignment distance equal to the initial alignment distance. In the absence of other external loads, the solution should immediately converge at time 1.0. The alignment ends at time 2.0, with the two nodes overlapping.

```
TIMESTEP
CLEAR
2 1.0
DATAEND
*
ALIGN-DISTANCE 1
ENTRIES TIME OPTION
CLEAR
1.0 TBFACTOR
2.0 ALIGNED
DATAEND
*
EGROUP ALIGNMENT 1 ALIGN-DISTANCE=1
EDATA
ENTRIES N1 N2
CLEAR
1 2
DATAEND
```

Example 3: This example is similar to example 2. However the alignment takes place over two solution steps, with the prescribed alignment distance at time 2 equal to the initial alignment distance, multiplied by 0.1.

```
TIMESTEP
CLEAR
3 1.0
DATAEND
*
ALIGN-DISTANCE 1
ENTRIES TIME OPTION FACTOR
CLEAR
1.0 TBFACTOR
```

```
2.0 TBFACOR 0.1
3.0 ALIGNED
DATAEND
*
EGROUP ALIGNMENT 1 ALIGN-DISTANCE=1
EDATA
ENTRIES N1 N2
CLEAR
1 2
DATAEND
```

Example 4: This example is similar to example 3. However, the alignment begins at solution time 2.0. The prescribed alignment distance at time 2.0 is equal to the alignment distance at time 1.0. Note that time 1.0 is the solution time that immediately precedes time 2.0. The prescribed alignment distance at time 3.0 is equal to the alignment distance at time 1.0, multiplied by 0.1.

```
TIMESTEP
CLEAR
4 1.0
DATAEND
*
ALIGN-DISTANCE 1
ENTRIES TIME OPTION FACTOR
CLEAR
2.0 TBFACOR
3.0 TBFACOR 0.1
4.0 ALIGNED
DATAEND
*
EGROUP ALIGNMENT 1 ALIGN-DISTANCE=1
EDATA
ENTRIES N1 N2
CLEAR
1 2
DATAEND
```

Example 5: This example is similar to example 4. However, the times specified in ALIGN-DISTANCE do not correspond to solution times. The alignment begins at solution time 3.0. The prescribed alignment distance at time 3.0 is obtained as follows:

Dummy alignment distance at time 2.1 = alignment distance at time 2.0 (2.0 is the solution time immediately preceding time 2.1)

Dummy alignment distance at time 3.1 = alignment distance at time 2.0, multiplied by 0.1.

Prescribed alignment distance at time 3.0 = Linear interpolation of dummy alignment distances at times 2.1, 3.1. In this case, the result is equal to the alignment distance at time 2.0, multiplied by 0.19, since

$$0.19 = 1.0 + (0.1 - 1.0) * (3.0 - 2.1) / (3.1 - 2.1)$$

The prescribed alignment distance at time 4.0 is obtained as follows:

Dummy alignment distance at time 3.1 = alignment distance at time 2.0, multiplied by 0.1

Dummy alignment distance at time 4.1 = zero

Prescribed alignment distance at time 4.0 = Linear interpolation of dummy alignment distance at times 3.1, 4.1.

The prescribed alignment distance at time 5.0 is equal to zero.

```
TIMESTEP
CLEAR
5 1.0
DATAEND
*
ALIGN-DISTANCE 1
ENTRIES TIME OPTION FACTOR
CLEAR
2.1 TBFACOR
3.1 TBFACOR 0.1
4.1 ALIGNED
DATAEND
*
EGROUP ALIGNMENT 1 ALIGN-DISTANCE=1
EDATA
ENTRIES N1 N2
CLEAR
1 2
DATAEND
```

Example 6: In this example, nodes 1 and 2 are aligned using a rotational alignment only. The alignment starts at time 3 and the

rotations are aligned at time 4. Here we suppose that the two nodes are each connected to Hermitian beam elements. Also, it is not necessary to set the triads in the EDATA command since there is only one triad per node.

The nodes are aligned when the b triad axes of the two nodes are coincident. Since no triadset is defined, the b triad axes correspond to the r-s-t axes of the attached beam elements.

```
ALIGN-ROTATION 1
ENTRIES TIME OPTION
CLEAR
3.0 TBFACOR
4.0 ALIGNED
DATAEND
*
EGROUP ALIGNMENT 1 ALIGN-ROTATION=1
EDATA
ENTRIES N1 N2
CLEAR
1 2
DATAEND
```

Example 7: In this example, nodes 1 and 2 are aligned using a rotational alignment only. The alignment starts at time 3 and the rotations are aligned at time 4.

Here we explicitly define the a and b triads at the nodes. Triadset 1 gives the triads for node 1 and triadset 2 gives the triads for node 2.

Triadset 1: The a triad initially has directions $a1=(0,-1,0)$, $a2=(-1,0,0)$ (components in the global system). $a3$ is computed to create a right-handed orthonormal system.

The b triad has offset (1, 2, 3). This offset is measured in the $a1, a2, a3$ directions. The b triad directions are $b1=(0,1,0)$, $b2=(0,0,1)$. $b3$ is computed to create a right-handed orthonormal system. Again, the components of the $b1, b2, b3$ axes are expressed in the a triad coordinate system.

Thus for demonstration, the initial b triad offset and directions in the global system can be calculated as follows:

$a3=(0,0,-1)$ in global system

b3=(1,0,0) in a triad system

offset=1*(0,-1,0) + 2*(-1,0,0) + 3*(0,0,1) = (-2,-1,3) in global system

b1=0*(0,-1,0) + 1*(-1,0,0) + 0*(0,0,-1) = (-1,0,0) in global system

b2=0*(0,-1,0) + 0*(-1,0,0) + 1*(0,0,-1) = (0,0,-1) in global system

b3=1*(0,-1,0) + 0*(-1,0,0) + 0*(0,0,-1) = (0,-1,0) in global system

Triadset 2: The a triad initially has directions a1=(1,0,0), a2=(0,0,1). a3 is computed to create a right-handed orthonormal system.

The b triad has offset (-1, -2, -3). The b triad directions are b1=(-1,0,0), b2=(0,1,0) (again, the components of b1, b2 are expressed in the a triad coordinate system). b3 is computed to create a right-handed orthonormal system.

The nodes are aligned when the b triad axes of the two nodes are coincident. The triad offsets are not used in this example.

```
TRIADSETS
ENTRIES,
  TRIADSET,
    AOPTION A1X  A1Y  A1Z  A2X  A2Y  A2Z,
    BOPTION B0A1 B0A2 B0A3,
            B1A1 B1A2 B1A3 B2A1 B2A2 B2A3
CLEAR
  1,
  VECTORS  0.0 -1.0  0.0 -1.0  0.0  0.0,
  VECTORS  1.0  2.0  3.0,
            0.0  1.0  0.0  0.0  0.0  1.0
  2,
  VECTORS  1.0  0.0  0.0  0.0  0.0  1.0,
  VECTORS -1.0 -2.0 -3.0,
            -1.0  0.0  0.0  0.0  1.0  0.0
DATAEND
SET TRIADSET NODES
CLEAR
1 1
2 2
DATAEND

ALIGN-ROTATION 1
ENTRIES TIME OPTION
CLEAR
```

```
3.0 TBFACTOR
4.0 ALIGNED
DATAEND
*
EGROUP ALIGNMENT 1 ALIGN-ROTATION=1
EDATA
ENTRIES N1 N2
CLEAR
1 2
DATAEND
```

Example 8: In this example, nodes 1 and 2 are aligned using both a translational and a rotational alignment. The same triadsets are used as in the previous example. However different triadset numbers are assigned to the nodes, just to show that the triadset number need not be the node number.

The translational alignment starts at time 1 and the translations are aligned at time 2. The nodes are translationally aligned when the offsets of the b triad axes of the two nodes are coincident.

The rotational alignment starts at time 3 and the rotations are aligned at time 4. The nodes are rotationally aligned when the b triad axes of the two nodes are coincident.

```
TRIADSETS
ENTRIES,
  TRIADSET,
    AOPTION A1X A1Y A1Z A2X A2Y A2Z,
    BOPTION B0A1 B0A2 B0A3,
           B1A1 B1A2 B1A3 B2A1 B2A2 B2A3
CLEAR
  10,
    VECTORS 0.0 -1.0 0.0 -1.0 0.0 0.0,
    VECTORS 1.0 2.0 3.0,
           0.0 1.0 0.0 0.0 0.0 1.0
  20,
    VECTORS 1.0 0.0 0.0 0.0 0.0 1.0,
    VECTORS -1.0 -2.0 -3.0,
           -1.0 0.0 0.0 0.0 1.0 0.0
DATAEND
SET TRIADSET NODES
CLEAR
1 10
2 20
DATAEND
```

```
ALIGN-TRANSLATION 1
ENTRIES TIME OPTION
CLEAR
1.0 TBFACOR
2.0 ALIGNED
DATAEND
*
ALIGN-ROTATION 1
ENTRIES TIME OPTION
CLEAR
3.0 TBFACOR
4.0 ALIGNED
DATAEND
*
EGROUP ALIGNMENT 1 ALIGN-TRANSLATION=1,
    ALIGN-ROTATION=1
EDATA
ENTRIES N1 N2
CLEAR
1 2
DATAEND
```

Example 9: In this example, nodes 1 and 2 are assumed to be at the end of a straight line of beam nodes. It is desired to bend the straight line so that the line becomes a circle.

The rotational alignment starts at time 1. At time 2, the prescribed rotational alignment is approximately 120 degrees with alignment axis (0,0,1). The components of the axis are in the c triad system of node 1. (For this problem, since no triadset is used, the a, b and c triads of node 1 all coincide, therefore the components of the axis are in the a triad system of node 1, in other words, the direction of the t axis at node 1.)

At time 3, the prescribed rotational alignment is approximately 240 degrees with alignment axis (0,0,1). At time 4, the prescribed alignment is set to 0. Since the rotational interpolation always takes the shortest angular distance, the prescribed alignment between times 3 and 4 increases from 240 degrees to 360 degrees.

The nodes are rotationally aligned when the b triads at node 1 and 2 have the same directions.

The change in prescribed rotational alignment is always less than

about 120 degrees. This change must be less than 180 degrees per time step.

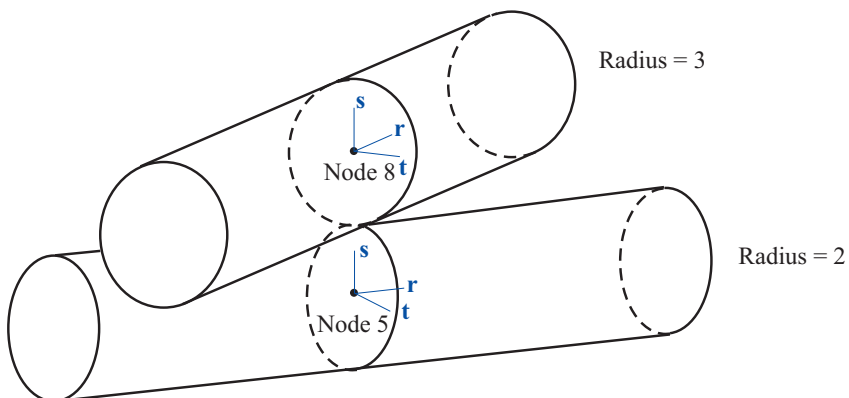
At time 5, the translational alignment starts and at time 6, the translational alignment ends. The nodes are translationally aligned when the nodes coincide.

```
ALIGN-TRANSLATION 1
ENTRIES TIME OPTION
CLEAR
5.0 TBFACOR
6.0 ALIGNED
DATAEND
*
ALIGN-ROTATION 1
ENTRIES TIME OPTION ANGLE AXIS1 AXIS2 AXIS3
CLEAR
1.0 TBFACOR
2.0 ANGLE 2.1 0.0 0.0 1.0
3.0 ANGLE 4.2 0.0 0.0 1.0
4.0 ALIGNED
DATAEND
*
EGROUP ALIGNMENT 1 ALIGN-TRANSLATION=1,
                  ALIGN-ROTATION=1

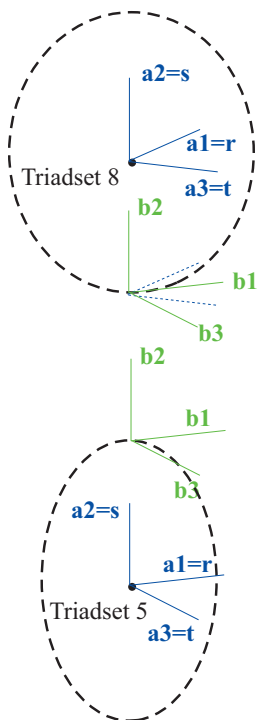
EDATA
ENTRIES N1 N2
CLEAR
1 2
DATAEND
```

Example 10: In this example, we want to simulate body-body contact.

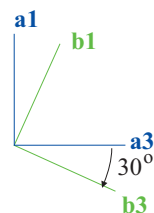
Fig 2.12-17 shows the bodies in the aligned configuration.



(a): At alignment



(b): Definition of triadsets



(c): Top view of triadset 8

Figure 2.12-17: Simulation of body-body contact

Triadset 5 is attached to node 5 and triadset 8 is attached to node 8.

The triadsets are defined so that the b triads are aligned in the aligned configuration:

a triads: The a triads are the same as the triads from the attached elements.

b triads, triadset 5: The b triads are offset from the node in the a2 direction:

$$(b0a1, b0a2, b0a3) = (0, 2, 0)$$

The b triad directions are the same as the a triad directions (the default)

$$(b1a1, b1a2, b1a3) = (1, 0, 0)$$

$$(b2a1, b2a2, b2a3) = (0, 1, 0)$$

$$(b3a1, b3a2, b3a3) = (0, 0, 1)$$

b triads, triadset 8: The b triads are offset from the node in the negative a2 direction:

$$(b0a1, b0a2, b0a3) = (0, -3, 0)$$

The b triad directions are rotated relative to the a triad directions:

$$(b1a1, b1a2, b1a3) = (\cos 30^\circ, 0, \sin 30^\circ)$$

$$(b2a1, b2a2, b2a3) = (0, 1, 0)$$

$$(b3a1, b3a2, b3a3) = (-\sin 30^\circ, 0, \cos 30^\circ)$$

c triads: The c triads are the same as the b triads.

The alignment element is defined using node 5 as local node 1 and node 8 as local node 2.

At alignment, we would like to have the b triad origins coincide. This means that we need high translational stiffnesses in all three coordinate directions.

We would also like to have the axis of rotation correspond to the $c2 = b2$ direction of node 5. Therefore we need high rotational stiffnesses in the $c1$ and $c3$ coordinate directions, but a low rotational stiffness in the $c2$ coordinate direction. In addition, we would like the stiffness in the $c2$ coordinate direction to be periodic. Thus, if node 5 is rotated 360 degrees relative to node 8, the alignment element forces are unchanged.

We will reach alignment by simultaneously applying the translational and rotational alignments in five time steps.

```
TRIADSETS
ENTRIES TRIADSET,
      AOPTION,
      BOPTION B0A1 B0A2 B0A3,
              B1A1 B1A2 B1A3 B2A1 B2A2 B2A3,
CLEAR
      5,
      ELEMENTS,
      VECTORS 0.0 2.0 0.0,
              1.0 0.0 0.0 0.0 1.0 0.0
      8,
      ELEMENTS,
      VECTORS 0.0 -3.0 0.0,
              0.86603 0.0 0.5 0.0 1.0 0.0
DATAEND
SET TRIADSET NODES
CLEAR
5 5
8 8
DATAEND

ALIGN-TRANSLATION 1 KTC1=1E15 KTC2=1E15 KTC3=1E15
ENTRIES TIME OPTION
CLEAR
1.0 TBFACTOR
6.0 ALIGNED
DATAEND
*
ALIGN-ROTATION 1 KRC1=1E15 KRC2=2 KRC3=1E15,
                KRC2TYPE=SAWTOOTH
ENTRIES TIME OPTION
CLEAR
1.0 TBFACTOR
6.0 ALIGNED
DATAEND
*
EGROUP ALIGNMENT 1 ALIGN-TRANSLATION=1,
                  ALIGN-ROTATION=1
ENODES
```

```
ENTRIES EL N1 N2
CLEAR
1 5 8
DATAEND
```

2.12.14 Element output

The element can either output nodal point forces or alignment data (the choice is made in the EGROU ALIGNMENT command).

When the element outputs alignment data, two levels of output can be selected:

Usual output: The actual (total) alignments are output. In the notation of Section 2.12.4 to 2.12.6:

Translation: $\mathbf{x}\mathbf{b}^2\mathbf{b}^1$ in component form:
 $(xb^2b^1_{c1}, xb^2b^1_{c2}, xb^2b^1_{c3})$. The postprocessing variables are
 ALIGN_TRANS_ACTUAL-C{123}.

Distance: $D = \text{length of vector } \mathbf{x}\mathbf{b}^2\mathbf{b}^1$. The postprocessing variable is ALIGN_DIST_ACTUAL.

Rotation: $\mathbf{V}\mathbf{B}^2\mathbf{B}^1$, transformed into an equivalent angle-axis representation, with the axis given in the c triad system:
 $\alpha (n_{c1}, n_{c2}, n_{c3})$. The angle α is output in degrees or radians, depending upon the setting of ALIGN-ROTATION UNITANGLE. The postprocessing variables are
 ALIGN_ROT_ACTUAL_MAGNITUDE,
 ALIGN_ROT_ACTUAL_AXIS-C{123}

Verbose output: In addition to the usual output, verbose output can be selected.

For printing of results, verbose output is selected either using

```
PRINTOUT PRINTDEFAULT=STRAINS
EGROUP ALIGNMENT PRINT=DEFAULT
EDATA ... printi=DEFAULT
```

or

```
EGROUP ALIGNMENT PRINT=VERBOSE  
EDATA ... print_i=DEFAULT
```

or

```
EDATA ... print_i=VERBOSE
```

For saving of results, verbose output is selected using EGROUP ALIGNMENT SAVE=VERBOSE.

Verbose output includes the usual output described above, and also the following additional output:

Translation:

$\mathbf{x}^1\mathbf{d}^1\mathbf{b}^1$ in component form: $(xd^1b^1_{c1}, xd^1b^1_{c2}, xd^1b^1_{c3})$. The postprocessing variables are ALIGN_TRANS_PRESC-C{123}.

$\mathbf{x}^2\mathbf{b}^1\mathbf{d}^1$ in component form: $(xb^2d^1_{c1}, xb^2d^1_{c2}, xb^2d^1_{c3})$. The postprocessing variables are ALIGN_TRANS_MISALIGN-C{123}.

The alignment forces (F_{c1}, F_{c2}, F_{c3}) . The postprocessing variables are ALIGN_TRANS_FORCE-C{123}.

Distance:

D^P . The postprocessing variable is ALIGN_DIST_PRESC.

$D - D^P$. The postprocessing variable is ALIGN_DIST_MISALIGN.

F_d . The postprocessing variable is ALIGN_DIST_FORCE.

Rotation:

$\mathbf{V}^1\mathbf{D}^1\mathbf{B}^1$, transformed into an equivalent angle-axis representation, with the axis given in the c triad system:

$\gamma^p (n_{c1^1}^p, n_{c2^1}^p, n_{c3^1}^p)$. The angle γ^p is output in degrees or radians, see above. The postprocessing variables are ALIGN_ROT_PRESC_MAGNITUDE, ALIGN_ROT_PRESC_AXIS-C{123}

$\mathbf{VB}^2\mathbf{D}^1$, transformed into an equivalent angle-axis representation, with the axis given in the c triad system: $\theta^m (n_{c1^1}^m, n_{c2^1}^m, n_{c3^1}^m)$. The angle θ^m is output in degrees or radians, see above. The postprocessing variables are ALIGN_ROT_MISALIGN_MAGNITUDE, ALIGN_ROT_MISALIGN_AXIS-C{123}

The alignment moments (M_{c1}, M_{c2}, M_{c3}) , output as a resultant moment about an axis $M (n_{c1^1}^M, n_{c2^1}^M, n_{c3^1}^M)$. The postprocessing variables are ALIGN_ROT_MOMENT_MAGNITUDE, ALIGN_ROT_MOMENT_AXIS-C{123}

2.12.15 Recommendations

- 1) When using the rotation alignment, it is best to define the b triads so that they have the same orientation at alignment.
- 2) When using the distance alignment, the directions of the b triads, and also the c triads, are not used.
- 3) When using the translation alignment, the directions of the b triads are not used.
- 4) If it is necessary to keep the translational misalignment small for all coordinate directions, then the stiffness in each translational direction should be set to a large value. In this case the directions of the c triads are not important and therefore the c triads can be set to the b triads (the default).
- 5) If it is necessary to keep the rotational misalignment small for all coordinate directions, then the stiffness in each rotational direction should be set to a large value. In this case the directions of the c

triads are not important and therefore the c triads can be set to the b triads (the default).

6) Hence the directions of the c triads only become important when the stiffnesses are different in the different coordinate directions, or when the translational components are explicitly specified, or when the rotation angle-axis is explicitly prescribed.

7) When defining the triadsets, it is easiest to consider the nodes in the aligned configuration, then define the triadsets in the aligned configuration.

8) The nonsymmetric solver can be useful when the direction of the prescribed rotational alignment changes during the analysis.

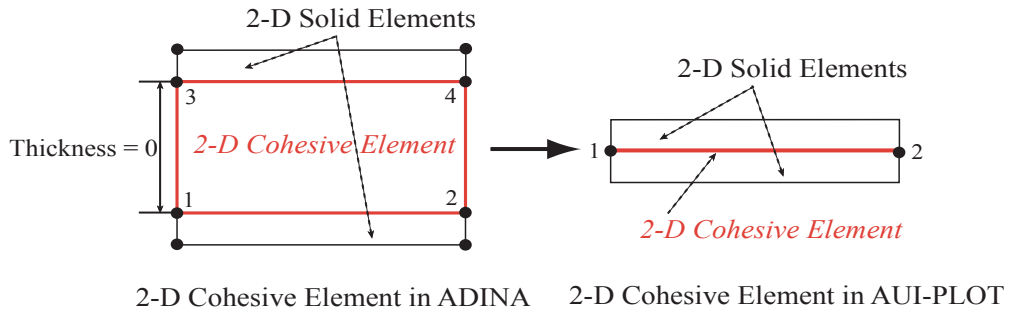
2.13 Cohesive elements

2.13.1 General considerations

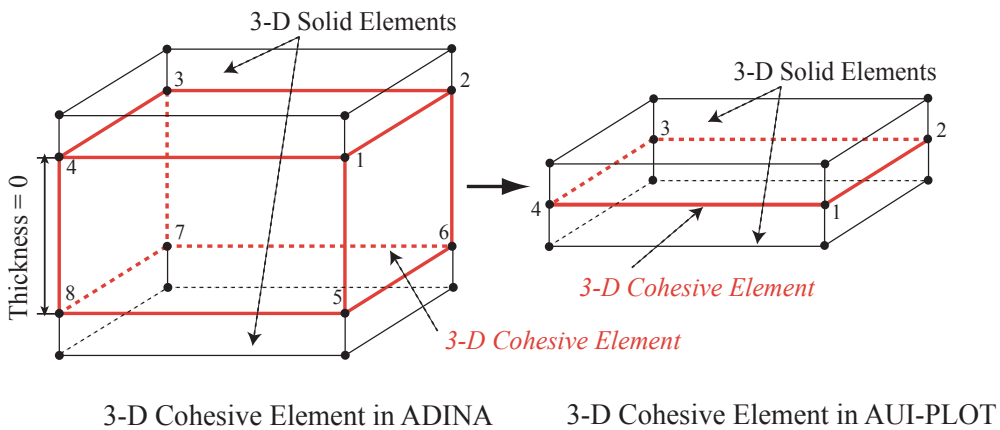
- The cohesive element is a 4-node (for 2-D) or 8-node (for 3-D) element that can be employed to model the separation at zero-thickness interfaces, such as laminated composites, bonded interfaces, etc.

2.13.2 Element formulation

- The cohesive element is composed of top and bottom surfaces with initially zero thickness. As shown in Figure 2.13-1, the 2-D cohesive element has 4 nodes that connect 2-D 4-node solid elements (axisymmetric, plane stress or plane strain elements). The 3-D cohesive element has 8 nodes that only connect 3-D 8-node solid elements.
- The mid-surface of the cohesive elements is tracked by averaging the coordinates of nodal pairs from the top and bottom surfaces so that the effect of geometrical nonlinearity can be taken into account.



(a) 4-node 2-D cohesive element



(b) 8-node 3-D cohesive element

Figure 2.13-1: 4-node and 8-node cohesive elements

2.13.3 Constitutive law of cohesive element

- The following is a quick summary of the basic concepts used in the constitutive law of the cohesive element. For further information, please see the following references:

ref. Turon, A., Camanho, P.P., Costa, J., and Davila, C.G., "A Damage Model for the Simulation of Delamination in Advanced Composites under Variable-Mode Loading,"

Mechanics of Materials, Vol. 38, No. 11, pp. 1072-1089, 2006.

ref. Camanho, P., Dávila, C., "Mixed-mode decohesion finite elements for the simulation of delamination in composite materials," *NASA/TM-2002-211737*, pp. 1-37, 2002.

ref. Alfano, G., Crisfield, M.A., "Finite element interface models for the delamination analysis of laminated composites: mechanical and computational issues," *International Journal for Numerical Methods in Engineering*, 50 (7), pp. 1701–1736, 2001.

- The cohesive element uses a bilinear constitutive law that relates the traction, τ , to the relative displacement, Δ , at the element mid-surface as shown in Figure 2.13-2. Initially linear elastic behavior followed by the initiation and evolution of damage is assumed. After the onset of damage, the stiffnesses of the cohesive element are gradually reduced to zero in its softening envelope. Unloading and reloading in the softening envelope are indicated by the arrows in Figure 2.13-2. d is a scalar damage variable which is output as "COHESIVE_DAMAGE" in the AUI.

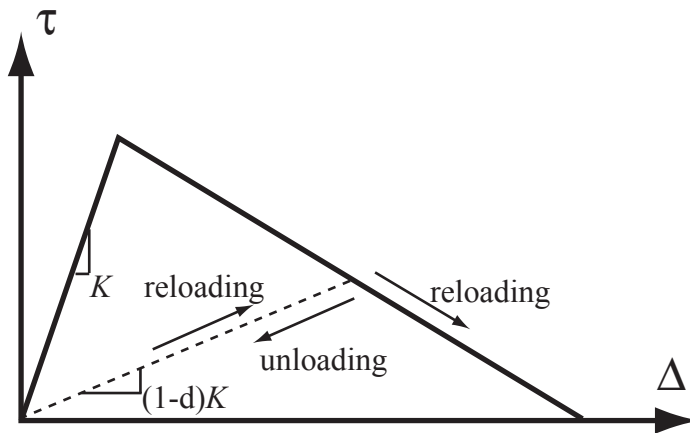


Figure 2.13-2 Bilinear constitutive law of cohesive element

- For pure mode I, II or III loading, after the interfacial normal or shear tractions reach their respective tensile or shear strengths, the stiffnesses are gradually reduced to zero. The area under the curve

is the respective (mode I, II or III) fracture toughness. The onset of damage at the interface can be determined by comparing the tractions with their respective allowable.

- Under mixed-mode loading, the onset of damage may occur before any of the traction components involved reach their respective allowable. In this case, the following criteria are provided in ADINA to predict delamination growth under mixed-mode loadings.

- ▶ Power law criterion
- ▶ Benzeggagh–Kenane (B-K) criterion

- The power law criterion is established in terms of a linear or quadratic interaction between the energy release rates using

$$\left(\frac{G_I}{G_{IC}} \right)^\alpha + \left(\frac{G_{II}}{G_{IIC}} \right)^\alpha = 1$$

where G_I and G_{II} are the mode I and mode II energy release rate; G_{IC} and G_{IIC} are the mode I and mode II fracture toughness; α is mixed-mode interaction parameter which is usually between 1 and 2.

- With Benzeggagh–Kenane (B-K) criterion, the critical energy release rate G_C is expressed as a function of the mode I and mode II fracture toughness and a mixed-mode interaction parameter η ,

$$G_C = G_{IC} + (G_{IIC} - G_{IC}) \left(\frac{G_S}{G_T} \right)^\eta$$

where G_T is total energy release rate under mixed-mode loading; G_S is the energy release rate for shear loading.

- The properties required to define the bilinear constitutive law are the mode I and mode II fracture toughness (G_{IC} , G_{IIC}), the penalty stiffness (K), the normal and shear interfacial strengths,

and the mixed-mode interaction criterion and its parameter. Please note that the same penalty stiffness is assumed for all three modes and the shear interfacial strengths in the two orthogonal directions are assumed to be the same.

2.13.4 Numerical integration

For the calculation of all element matrices and vectors, numerical Newton-Cotes integration is used. As shown in Figure 2.13-3, 2×1 Newton-Cotes integration points are used for 2-D 4-node cohesive element, and 2×2 Newton-Cotes integration points for 3-D 8-node cohesive element.

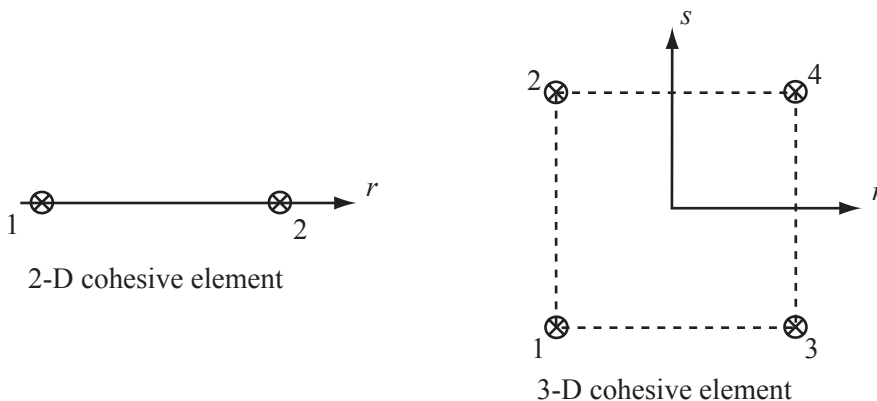


Figure 2.13-3: Newton-Cotes integration points layout

2.13.5 Element output

You can request that ADINA either print or save stresses/strains.

Stresses/strains: Each element outputs, at its nodal points, the following information to the porthole file. This information is accessible in the AUI using the given variable names.

COHESIVE_NORMAL_STRESS,
 COHESIVE_TANGENTIAL_STRESS,
 COHESIVE_NORMAL_STRAIN,
 COHESIVE_TANGENTIAL_STRAIN, COHESIVE_DAMAGE

2.14 Connector elements

2.14.1 Overview

The connector element is a two-node generalized spring/damper element that can undergo large rotations and translations. The connector element calculates forces and/or moments from the translational and/or rotational motions of its nodes relative to the element's undeformed state and its reference triad, each of which can be defined by the user to suit a given modeling purpose.

Possible applications for the connector element include:

- ▶ Any modeling situation requiring a linear or nonlinear spring/damper which can undergo large rotations with individual properties for each degree of freedom which convect with the element's local coordinate system
- ▶ Modeling mechanical connections such as flexible bushings, bearings, or sliders
- ▶ Simulations involving networks or "spiders" of linkages with arbitrarily defined elastic and damping properties
- ▶ An inexpensive 2-node element representation of a complex (and potentially nonsymmetric) linear or multilinear elastic and/or damping response of a connecting structure which can undergo large rotations

An important defining feature of the connector element is its convecting local coordinate system such that for rigid body motions, the relative nodal positions and orientations remain unchanged and the element's internal forces transform with the rigid body motions.

Connector elements are always geometrically nonlinear. There are three subtypes of connector elements

- ▶ Joint connector
- ▶ Matrix connector
- ▶ Multilinear matrix connector

The three connector element subtypes denote the input required. Matrix type connector elements are always materially linear.

However, the joint connector subtype can either be materially linear or nonlinear. Multilinear matrix connector elements have stiffness and damping matrices defined as multilinear function of relative displacement, relative velocity, elastic force, damping force, or temperature.

Connector element properties can be scaled as a function of time. Thus, depending on how the user defines the element's nodal triads and undeformed state, the connector element can generate internal forces to resemble a prescribed load or constraint.

2.14.1.1 Joint connector elements

Joint type connector elements take as input constant stiffness and/or damping coefficients or multilinear force-relative displacement/velocity responses for each of six degrees of freedom. For joint type connector elements, the user supplies the response in terms of its generalized (local) coordinates.

2.14.1.2 Matrix connector elements

Matrix type connector elements are useful for modeling the symmetric or nonsymmetric linear elastic and/or damping response of an arbitrary connecting structure which can undergo large translations and rotations. Rather than use a fully-3D model of the structure, the user can employ the matrix type connector element to substitute an inexpensive 2-node element with the appropriate structural response. The response convects properly with the element as it undergoes arbitrary motions.

The user may enter directly the stiffness and/or damping matrices which relate the local forces and moments to the element's translational and rotational motions, relative the element's local reference coordinate system. The user must input a full 12x12 stiffness (and/or damping) matrix for the nonsymmetric case or the upper triangular portions for the symmetric case.

Note: *For a detailed demonstration of obtaining the stiffness matrix for a structure and constructing an equivalent connector*

*element model, see **Primer Problems 67 and 68.***

It is, in general, possible for the user to enter a stiffness matrix that results in forces being generated under rigid body motions. This is known as a “non-equilibrium” matrix. By default, however, the program ignores any portion of the matrices that will cause the element to develop forces under rigid body motions.

2.14.1.3 Multilinear matrix connector elements

Multilinear matrix type connector elements extend the capabilities of the matrix connector and allow modeling the symmetric or nonsymmetric multilinear elastic and/or damping response of an arbitrary connecting structure. Rather than possessing a constant response, a multilinear matrix connector element’s response can be defined as a function of relative displacement, relative velocity, elastic force, damping force, or temperature.

The user may enter directly a series of stiffness and/or damping matrices and reference them as functions of relative displacement, relative velocity, elastic force, damping force, or temperature. Matrix responses are linearly interpolated between discrete entries.

As with the matrix connector element, each matrix relates the local forces and moments to the element’s translational and rotational motions, relative the element’s local reference coordinate system. The user must input full 12x12 stiffness (and/or damping) matrices for the nonsymmetric responses or the upper triangular portions for the symmetric responses. A multilinear matrix connector element’s response may include symmetric and nonsymmetric matrices.

2.14.1.4 Nodal triads

The connector element’s local nodes have an attached set of orthogonal base vectors known as triads. These triads translate and rotate with each local node and determine the nodes’ relative positions and orientations. The user can specify the triad orientations in a variety of ways and, in so doing, specify the initial state of the connector element to suit a given modeling requirement.

2.14.1.5 Reference axes

The user can select how the connector element's local convecting coordinate system is defined. The reference axes can either correspond to the axes defined by the triad at local node 1, or the reference axes can be taken as the average of the triads at local nodes 1 and 2. Of course, these triads and the reference systems can rotate in space during an analysis.

2.14.1.6 Undeformed configurations

The user can select one of three different ways of defining a connector element's undeformed configuration. In addition to initially placing nodes at specific locations relative to each other and the reference axes, the user can specify that the original configuration to be the undeformed configuration, or if the element is undeformed when its local nodes are coincident. Additionally, the user can specify that the element be undeformed when its nodes are at the projections of their original locations onto the element's local 1-direction. This ability to specify undeformed configurations is most useful for linear joint connector elements and matrix connector elements which always have constant responses with zero forces generated at the undeformed configuration.

2.14.2 Theory

2.14.2.1 Triads

Triads are sets of orthonormal base vectors. The connector element uses triads to describe rigidly translating and rotating frames which may then, in turn, be used to define relative motions (both translations and rotations) and describe a coordinate system for defining local convecting properties and forces.

Connector elements have two *a-triads* (one for each local node) and a single *b-triad*. The user specifies a-triads at each local node via the `CONN-DEFINE TRIAD1OPT` and `TRIAD2OPT` parameters.

The `EGROUP CONNECTOR REFERENCE` parameter then controls how the a-triads are used to determine the element's b-triad. The b-triad is used as the connector element's reference system to define the element properties and measure the relative displacements and velocities and, ultimately, to compute the internal forces and moments.

A-triads

Each node of the connector element has an attached triad. This triad is called the "a-triad," and the coordinate axes of the a-triad are denoted $\mathbf{a1}^n$, $\mathbf{a2}^n$, and $\mathbf{a3}^n$ for the n^{th} local node (the connector element always has 2 nodes). This triad is an orthonormal, right-handed coordinate system, with origin at the node. As the node translates and rotates, the triad rigidly translates and rotates with the node. Fig. 2.14-1 shows a 2D representation of the a-triads at each local node.

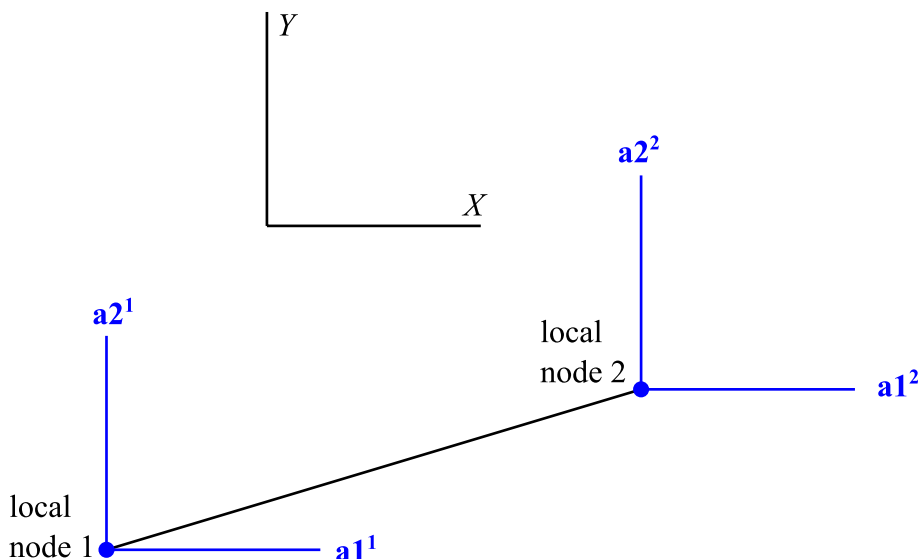


Figure 2.14-1: Connector element local nodes 1 and 2 with corresponding a-triads, in blue. In this case, both a-triads are aligned with the global X - and Y -axes (shown in black).

B-triad

Each connector element also has a reference triad, known as the b-triad. The connector element's properties are defined with respect to the b-triad, and the relative local displacements and velocities are measured with respect to the b-triad. The coordinate axes of the b-triad are denoted **b1**, **b2**, and **b3**.

The b-triad is an orthonormal right-handed coordinate system, with its origin either at local node 1 or midway between local nodes 1 and 2, depending on the option selected by `EGROUP CONNECTOR REFERENCE`. When `REFERENCE` is set to `AXES1`, the b-triad is the a-triad at node 1. When `REFERENCE` is set to `AXESAVG`, the orientation of the b-triad is averaged from the a-triads at local nodes 1 and 2, and the origin of the b-triad is midway between local nodes 1 and 2. Fig. 2.14-2 illustrates the b-triad when `REFERENCE = AXESAVG`.

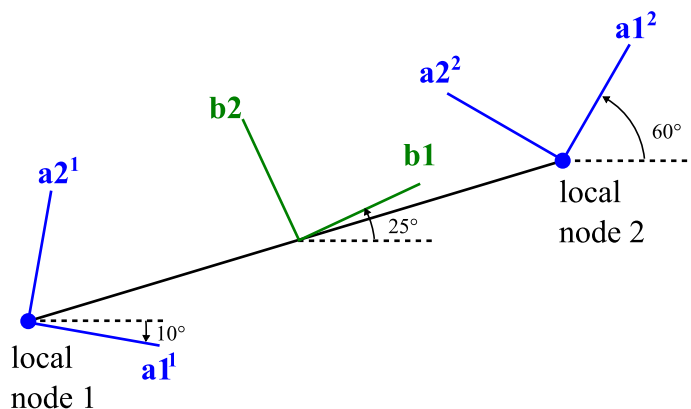


Figure 2.14-2: Illustrative 2D example showing the orientation of a connector element's b-triad (in green) when `REFERENCE = AXESAVG`. Note the orientation of the b-triad; it is rotated 25 degrees from the horizontal (dashed line), which is the orientation obtained by averaging the a-triads at nodes 1 and 2. The orientation of the b-triad depends only upon the orientations of the a-triads – it does not depend upon the locations of local nodes 1 and 2.

In either case, the b-triad translates and rotates as the element's nodes translate and rotate in space. The relation between nodal motions and reference axis orientation depends upon the `REFERENCE` option. In all cases, the element's properties convect with the reference axes.

It is possible to view the triads in AUI by selecting “Display Local System Triad” in the element depiction dialogue, or by using the AUI command `ELDEPICTION TRIAD=YES`. Fig. 2.14-3 shows the triad depictions. The b-triad is displayed separately midway between the nodes only when `REFERENCE=AXESAVG`.

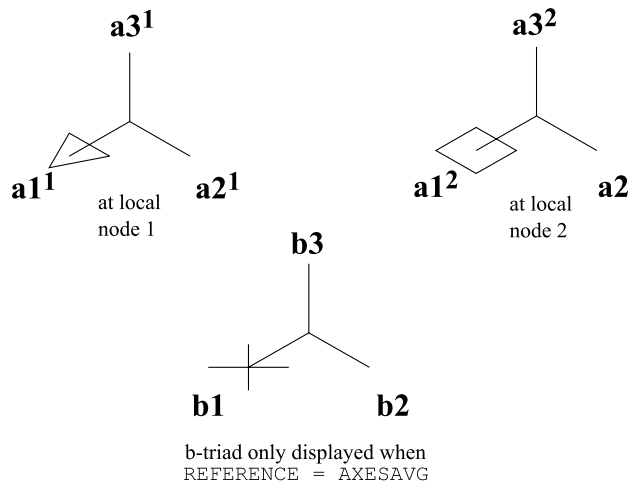


Figure 2.14-3: Connector element triad depictions in AUI when enabling “Display Local System Triad.”

In the following sections, the b-triad is not drawn when `REFERENCE=AXES1`. This is because the b-triad is the same as the a-triad at node 1 when `REFERENCE=AXES1`.

2.14.2.2 Undeformed configurations and local displacements and rotations

In the connector element, the undeformed configuration is defined by the relative positions of local nodes 1 and 2 for which no translational internal forces develop.

Consider the original state of the connector element shown in Fig. 2.14-4. That element is defined with `REFERENCE = AXES1`, which means that all local displacements and rotations are measured using the a-triad at local node 1. Local node 2 is originally located at local coordinates (L_1^O, L_2^O, L_3^O) , as measured in the b-triad coordinate system, with the “O” superscript denoting original local coordinates.

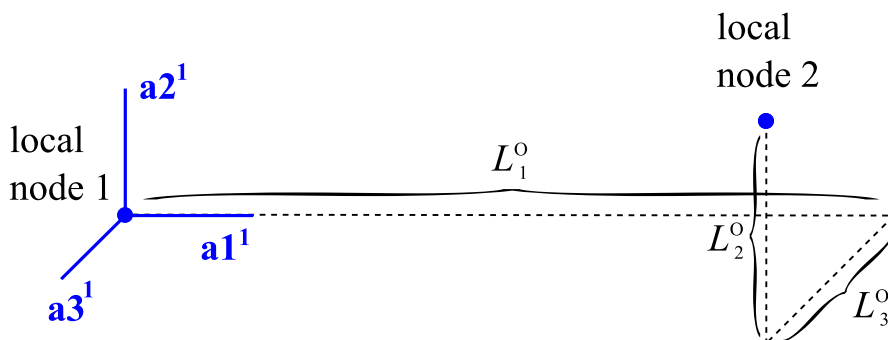


Figure 2.14-4: Original location of local node 2 with respect to the element's local origin when `REFERENCE = AXES1`. The original location of local node 2, with respect to the reference axes at node 1 is (L_1^O, L_2^O, L_3^O) . This figure shows the original configuration of the connector elements shown in Figs. 2.14-5 through 2.14-7, below.

The relationship between the local displacements of node 2 relative to node 1 z_i , current translational coordinates l_i , and undeformed translational coordinates L_i is

$$z_i = l_i - L_i,$$

where subscript $i = 1, 2, 3$ denotes components along the **b1**, **b2**, and **b3** directions, respectively. The connector element's local translational forces depend upon the local displacements z_i , as discussed in detail below.

The relationship between the original configuration and the undeformed configuration is specified using `CONN-DEFINE UNDEFORMED`. The choices are `ORIGINAL/LOCALORIGIN/B1PROJECT`.

Note well that these options only refer to the element's internal translational degrees of freedom. Relative rotations are always computed using the a-triads at local nodes 1 and 2 with respect to the current reference axes. Local rotations z_4 , z_5 , and z_6 are the relative rotations of node 2 with respect to node 1 about the local **b1**, **b2**, and **b3** axes, respectively.

UNDEFORMED = ORIGINAL

When `UNDEFORMED = ORIGINAL`, the connector element is undeformed when its local nodes are in their original locations, as shown in Fig. 2.14-5. That is, $L_i = L_i^O$. Thus, the element is undeformed ($z_i = 0, i = 1, 2, 3$) when $l_i = L_i^O$. This is the default selection.

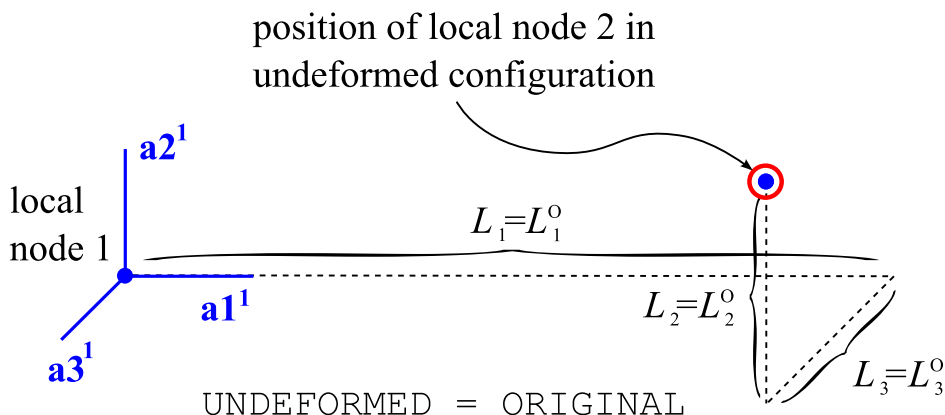


Figure 2.14-5: Important geometric quantities when `UNDEFORMED = ORIGINAL` and `REFERENCE = AXES1`. The location of local node 2, with respect to the reference axes at node 1 in the element's undeformed configuration is $(L_1 = L_1^0, L_2 = L_2^0, L_3 = L_3^0)$ as shown by the open red circle. In the element's original configuration, the local translational displacements z_1 , z_2 , and z_3 , are all zero.

UNDEFORMED = LOCALORIGIN

When `UNDEFORMED = LOCALORIGIN`, the connector element is undeformed when its local nodes are coincident, as shown in Fig. 2.14-6. That is, $L_i = 0$. Thus, the element is undeformed when l_1 , l_2 , and l_3 are zero. If the connector element in Fig. 2.14-6 were to remain in its original configuration, the original coordinates L_1^0 , L_2^0 , and L_3^0 would be interpreted as local translational displacements z_1 , z_2 , and z_3 .

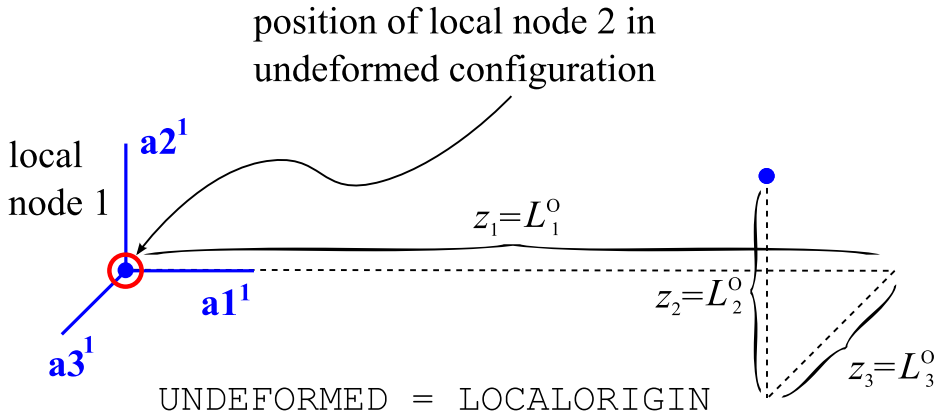


Figure 2.14-6: Important geometric quantities when $\text{UNDEFORMED} = \text{LOCALORIGIN}$ and $\text{REFERENCE} = \text{AXES1}$. The location of local node 2, with respect to the reference axes at node 1 in the element's undeformed configuration is $(L_1 = 0, L_2 = 0, L_3 = 0)$ as shown by the open red circle. In the element's original configuration, the quantities L_1^O , L_2^O , and L_3^O are interpreted as local translational displacements z_1 , z_2 , and z_3 , respectively.

UNDEFORMED = B1PROJECT

When $\text{UNDEFORMED} = \text{B1PROJECT}$, the connector element is undeformed when its local nodes are at the projections of their original positions onto the local **b1** axis, as shown in Fig. 2.14-7. That is, $L_1 = L_1^O$, and L_2 and L_3 are zero. Thus, the element is undeformed when $l_1 = L_1^O$ and l_2 and l_3 are zero. If the connector element in Fig. 2.14-7 were to remain in its original configuration, the original coordinates L_2^O and L_3^O would be interpreted as local translational displacements z_2 and z_3 , respectively.

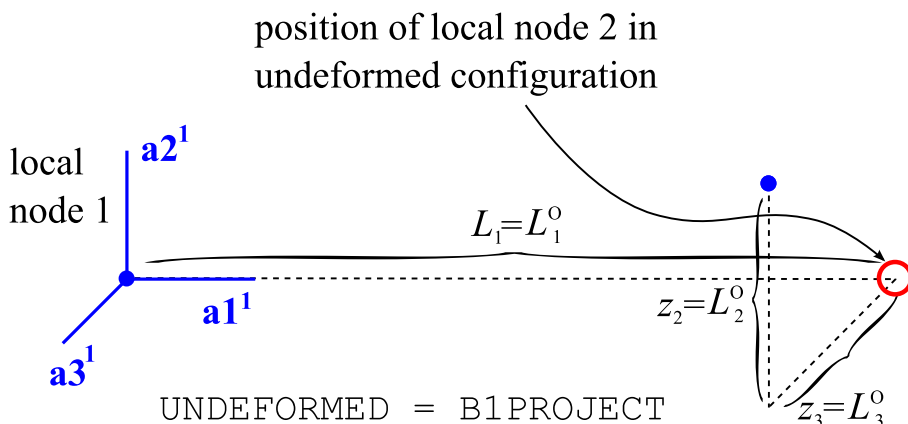


Figure 2.14-7: Important geometric quantities when UNDEFORMED = B1PROJECT and REFERENCE = AXES1. The location of local node 2, with respect to the reference axes at node 1 in the element's undeformed configuration is $(L_1 = L_1^0, L_2 = 0, L_3 = 0)$ as shown by the open red circle. In the element's original configuration, the quantities L_2^0 and L_3^0 are interpreted as local translational displacements z_2 and z_3 , respectively. Recall that when REFERENCE = AXES1, the local **b1** direction corresponds to the **a1**¹ direction.

2.14.2.3 Local forces and moments for joint type connector element

By default, joint connector elements compute the internal displacements and rotations from the motions of local node 2 relative to the triad at local node 1 (REFERENCE=AXES1). Given those local displacements z_1 , z_2 , z_3 and rotations z_4 , z_5 , z_6 of local node 2 relative to local node 1 in the connector element's reference system, the conjugate force or moment corresponding to the i^{th} degree of freedom for joint connector elements is

$$F_{z_i} = f_{z_i}(z_i) + g_{z_i}(\dot{z}_i),$$

where $f_{z_i}(z_i)$ and $g_{z_i}(\dot{z}_i)$ are the user-defined (linear or nonlinear) force-displacement and force-velocity responses, respectively.

2.14.2.4 Local forces and moments for matrix type connector element

By default, matrix and multilinear matrix connector elements compute internal displacements and rotations by averaging the motions of the local nodes (REFERENCE=AXESAVG). Consider the incremental displacements $\Delta u_1^i, \Delta u_2^i, \Delta u_3^i$ and rotations $\Delta \theta_1^i, \Delta \theta_2^i, \Delta \theta_3^i$ for local node i . The relationship between the 12 nodal displacements and rotations and the connector element's 6 local displacements is given by

$$\begin{bmatrix} \Delta u_1^1 \\ \Delta u_2^1 \\ \Delta u_3^1 \\ \Delta \theta_1^1 \\ \Delta \theta_2^1 \\ \Delta \theta_3^1 \\ \Delta u_1^2 \\ \Delta u_2^2 \\ \Delta u_3^2 \\ \Delta \theta_1^2 \\ \Delta \theta_2^2 \\ \Delta \theta_3^2 \end{bmatrix} = \begin{bmatrix} & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ \cdots & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \end{bmatrix} \begin{bmatrix} \Delta z_1 \\ \Delta z_2 \\ \Delta z_3 \\ \Delta z_4 \\ \Delta z_5 \\ \Delta z_6 \end{bmatrix},$$

where \mathbf{I} is the 6x6 identity matrix, and $\Delta z_1, \Delta z_2, \Delta z_3$ and $\Delta z_4, \Delta z_5, \Delta z_6$ are the element's local incremental displacements and rotations, respectively. These local displacements and rotations are also known as generalized coordinates. Thus, the conjugate force or

moment corresponding to the i^{th} degree of freedom for matrix connector elements is

$$F_{z_i} = K_{z_i z_j} z_j + C_{z_i z_j} \dot{z}_j,$$

where

$$K_{z_i z_j} = \frac{1}{4} (KAA_{ij} - KAB_{ij} - KBA_{ij} + KBB_{ij}),$$

and KAA , KAB , KBA , and KBB are the four 6x6 submatrices taken from the user-defined stiffness matrix where

$$\begin{bmatrix} F_1^1 \\ F_2^1 \\ F_3^1 \\ M_1^1 \\ M_2^1 \\ M_3^1 \\ F_1^2 \\ F_2^2 \\ F_3^2 \\ M_1^2 \\ M_2^2 \\ M_3^2 \end{bmatrix} = \begin{bmatrix} & & & & & & & & & & & \\ & & & & & & & & & & & \\ & & & & & & & & & & & \\ & & & & & & & & & & & \\ & & & & & & & & & & & \\ & & & & & & & & & & & \\ & & & & & & & & & & & \\ & & & & & & & & & & & \\ & & & & & & & & & & & \\ & & & & & & & & & & & \\ & & & & & & & & & & & \\ & & & & & & & & & & & \end{bmatrix} \begin{bmatrix} u_1^1 \\ u_2^1 \\ u_3^1 \\ \theta_1^1 \\ \theta_2^1 \\ \theta_3^1 \\ u_1^2 \\ u_2^2 \\ u_3^2 \\ \theta_1^2 \\ \theta_2^2 \\ \theta_3^2 \end{bmatrix},$$

and where F_1^i, F_2^i, F_3^i and M_1^i, M_2^i, M_3^i are the consistent nodal forces and moments, respectively, at local node i .

$C_{z_i z_j}$ is computed in the same manner.

2.14.2.5 Local forces and moments for multilinear matrix type connector element

Determination of $K_{z_i z_j}$ from user-defined, displacement-, velocity-, or temperature-dependent stiffness matrix $K_{z_i z_j}(z)$, $K_{z_i z_j}(\dot{z})$, or $K_{z_i z_j}(T)$

When the stiffness matrices are functions of local displacements, velocities, or temperature, the program execute a lookup/interpolation for the stiffness matrix for the given quantity. The stiffness matrix components are then interpolated between the discrete, user-provided matrices.

Once the stiffness matrix is obtained, the elastic force vector is obtained by using ${}^{t+\Delta t}F_{z_i}^e = {}^tF_{z_i}^e + K_{z_i z_j}(\bullet) \cdot ({}^{t+\Delta t}z_j - {}^tz_j)$.

Formulas for $C_{z_i z_j}$ are similar to formulas for $K_{z_i z_j}$, and the damping forces are computed in the same way.

Determination of $K_{z_i z_j}$ from user-defined generalized elastic force-dependent stiffness matrix $K_{z_i z_j}(f^e)$.

In this case, the stiffness matrix is defined in terms of a scalar $K_{z_i z_j} = K_{z_i z_j}({}^{t+\Delta t}\beta)$, which is, in turn, a function of the current force vector ${}^{t+\Delta t}\beta = \beta({}^{t+\Delta t}F_{z_i}^e)$. For example, if MATRIX-NL-K TYPE=ELASTIC-1 (see section 2.14.3.4), then ${}^{t+\Delta t}\beta = {}^{t+\Delta t}F_{z_1}^e$.

Thus, it is necessary for the program to implicitly determine the stiffness matrix which satisfies

$${}^{t+\Delta t}F_{z_i}^e = {}^tF_{z_i}^e + K_{z_i z_j}({}^{t+\Delta t}F_{z_j}^e) \cdot ({}^{t+\Delta t}z_j - {}^tz_j).$$

Formulas for $C_{z_i z_j}$ are similar to formulas for $K_{z_i z_j}$, and the damping forces are computed in the same way.

2.14.2.6 Consistent nodal point forces and moments

Consistent nodal point forces and moments are computed using the principle of virtual work, and the virtual work due to local forces and local displacements must equal the virtual work due to nodal point forces/moments and nodal point displacements.

$$F_{z_i} \delta z_i = \sum_{j=1,3} \left(F_j^1 \delta u_j^1 + M_j^1 \delta \theta_j^1 + F_j^2 \delta u_j^2 + M_j^2 \delta \theta_j^2 \right),$$

where subscripts $i = 1, \dots, 6$ correspond to the local degrees of freedom, and $j = 1, \dots, 3$ correspond to the global directions.

It is important to note that the local conjugate forces depend only upon the local nodal displacement, rotations and the element's local properties. Consequently, it is possible for a connector element to develop no local moments but to have consistent forces and moments develop at the nodes.

2.14.2.7 Illustrative examples

To provide more concrete demonstrations of the important concepts introduced above, we present some illustrative examples. In all the following examples, the connector element is shown in its initial state as defined by the user (at the top of the figure) and in a subsequent state (center of the figure).

The connector element's local forces are shown separately in an inset at the bottom of each figure. Fig. 2.14-8 describes how the local forces are shown. Consistent nodal point forces and moments are shown schematically acting on the local nodes.

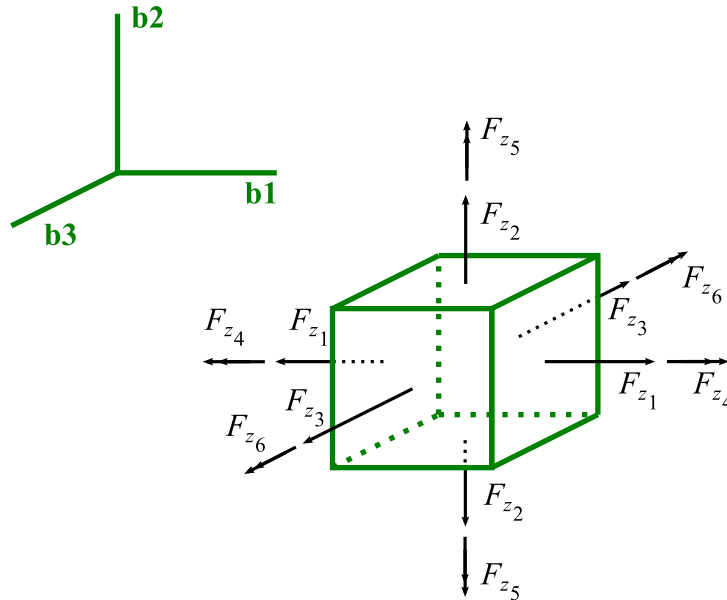


Figure 2.14-8: Schema showing the local forces and moments associated with a connector element. All forces and moments are shown here acting in the positive direction and are aligned with the reference axes (b-triad).

Note that although the initial state may represent a deformed state (as for Illustrative examples 5, 6, and 7), no forces are shown for the initial state. Also, for our purposes, it suffices to consider purely elastic cases.

Illustrative example 1

Joint type connector element with relative “axial” nodal displacements without rotations: As a simple example, consider a connector element defined with its a-triads initially aligned as shown in Fig. 2.14-9. The axes of each triad are parallel, and the reference axes are defined as the a-triad at node 1 (REFERENCE=AXES1). Node 2 moves relative to node 1 in the local 1-direction. In this case, the connector element will develop an internal force in the local 1-direction. The consistent nodal forces at each node are shown as \mathbf{F} .

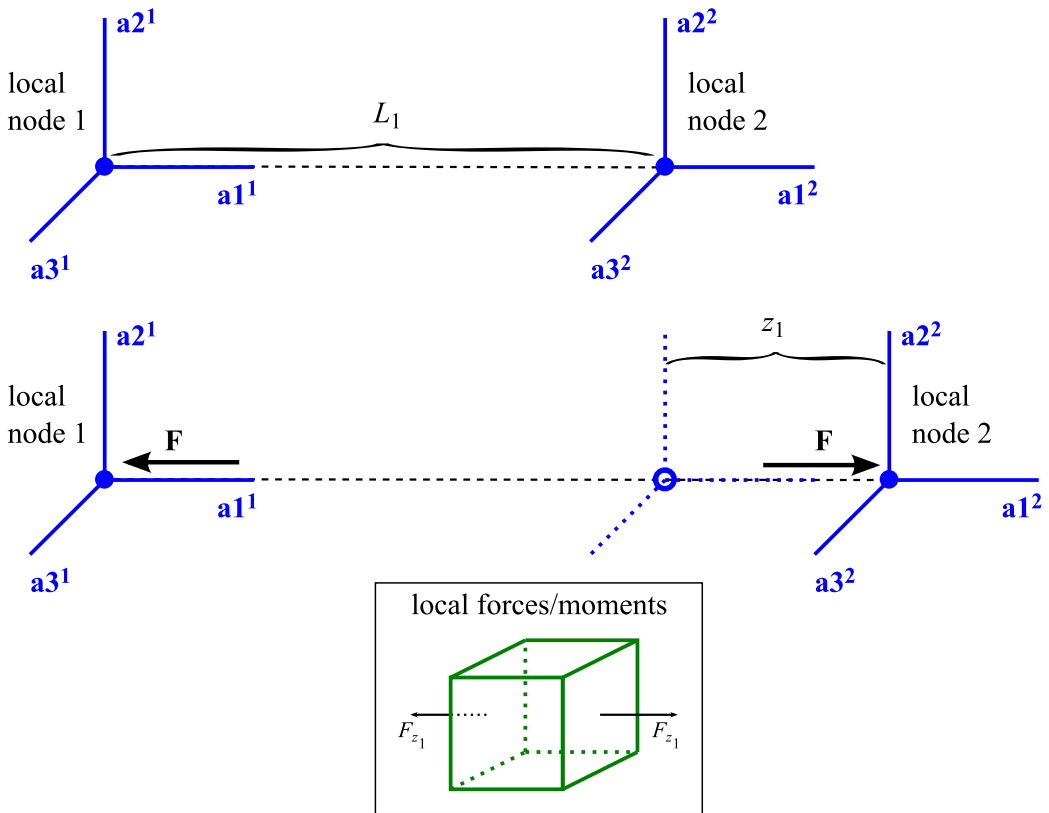


Figure 2.14-9: Illustrative example 1. Joint type connector element with REFERENCE=AXES1 and local node 2 undergoing a translation z_1 in the local $a1$ direction. This element will generate a conjugate local force F_{z_1} , as shown. If this element were to undergo rigid-body motions, the local force would remain unchanged.

Illustrative example 2

Joint type connector element with relative “transverse” nodal displacements without rotations: As a simple example, consider a connector element defined with its a-triads initially aligned as shown in Fig. 2.14-10. The axes of each triad are parallel, and the reference axes are defined as the a-triad at node 1 (REFERENCE=AXES1). Node 2 moves relative to node 1 in the local negative 2-direction, and the nodes do not undergo any relative rotations. In this case, the connector element will develop an internal force in the local 2-direction. The consistent nodal forces and moments at each node are shown as **F** and **M**. Note that although the connector element develops no internal moments, consistent nodal point moments will arise to satisfy moment equilibrium.

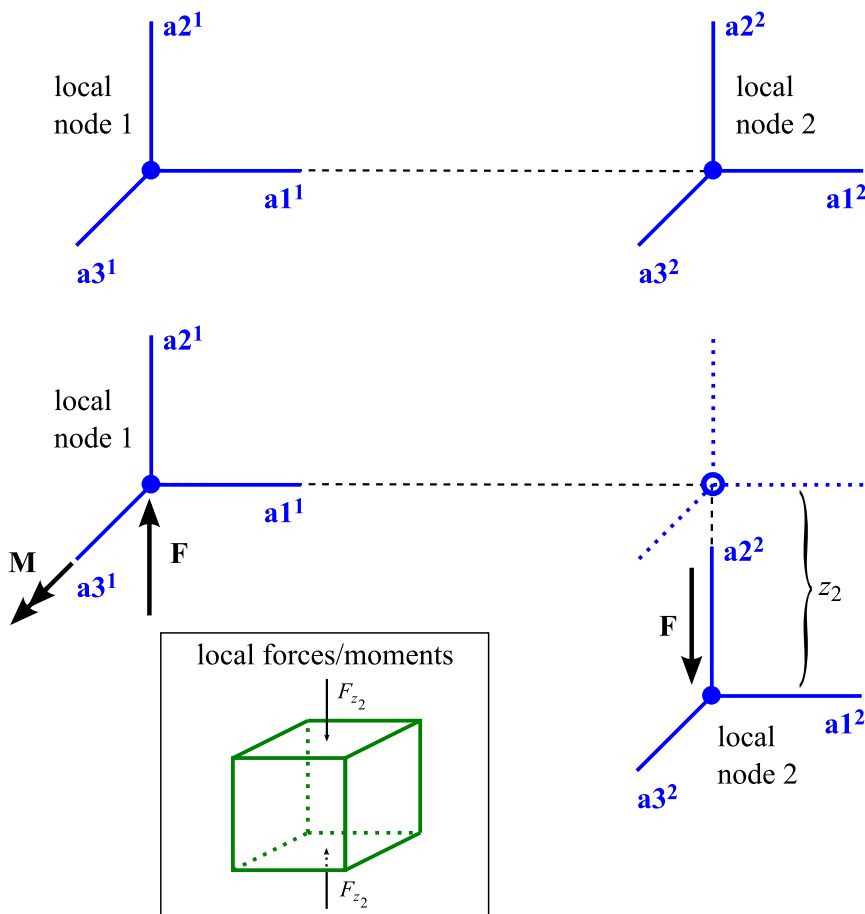


Figure 2.14-10: Illustrative example 2. Joint type connector element with `REFERENCE=AXES1` and local node 2 undergoing a negative displacement z_2 . This element will generate a conjugate local force F_{z_2} , as shown.

It may surprise the reader to see that a consistent nodal moment \mathbf{M} arises at local node 1 but no consistent nodal force moment arises at local node 2. To understand why this occurs, it is important to recall the principle of virtual work and consider variations in displacements about the element's final state.

Here, we can consider the principle of virtual work for planar motions

$$F_{z_2} \delta z_2 = F_1^1 \delta u_1^1 + F_2^1 \delta u_2^1 + M_3^1 \delta \theta_3^1 + F_1^2 \delta u_1^2 + F_2^2 \delta u_2^2 + M_3^2 \delta \theta_3^2$$

Setting each nodal point variation nonzero in sequence reveals the corresponding nodal point force or moment. For example, by setting $\delta \theta_3^1 \neq 0$ only, we obtain the equation $F_{z_2} \delta z_2 = M_3^1 \delta \theta_3^1$, and what remains is to determine how $\delta \theta_3^1$ affects δz_2 in the final configuration. Fig. 2.14-11 conceptualizes how variations in θ_3^1 affect variations in z_2 .

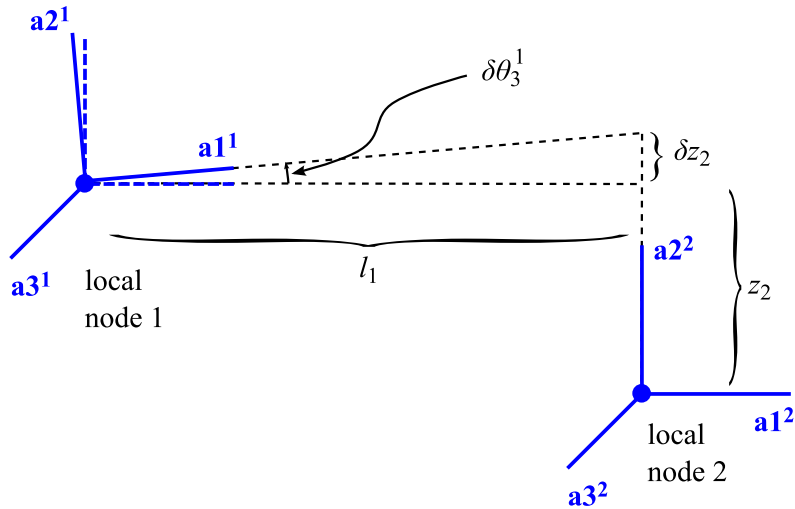


Figure 2.14-11: Conceptualization of how variations in θ_3^1 affect variations in z_2 about the connector element's final configuration. Due to how the generalized coordinates are defined for connector elements when REFERENCE=AXES1, it is clear that $\delta z_2 \cong l_1 \delta \theta_3^1$.

Setting $\delta\theta_3^2 \neq 0$ only, we now obtain the equation

$F_{z_2} \delta z_2 = M_3^2 \delta\theta_3^2$. Fig. 2.14-12 conceptualizes how variations in θ_3^2 affect variations in z_2 about the connector element's final configuration.

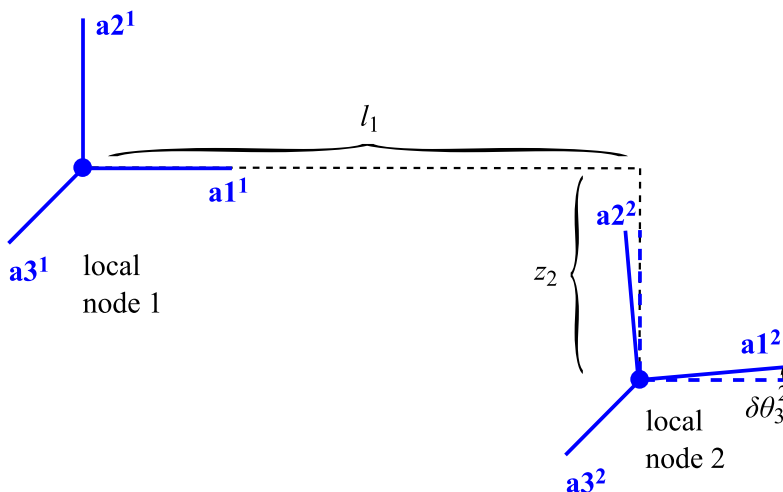


Figure 2.14-12: Conceptualization of how variations in θ_3^2 affect variations in z_2 about the connector element's final configuration. Due to how the generalized coordinates are defined for connector elements when REFERENCE=AXES1, $\delta\theta_3^2$ never affects δz_2 ,

Due to the way in which the connector element is defined when REFERENCE=AXES1, variations in θ_3^1 will have an effect on variations in z_2 , but variations in θ_3^2 will not have an effect on variations in z_2 . Thus, there can be no consistent nodal moment acting on local node 2 when the connector element is in the final configuration shown in Fig. 2.14-10.

Illustrative example 3

Joint type connector element with local node 2 undergoing rotations relative to local node 1: Consider the case of a connector element with node 2 undergoing a rotation relative to node 1, but no translations with respect to the local coordinate system. The reference axes are defined as the a-triad at node 1 (REFERENCE=AXES1). This element will generate an internal moment, as shown in Fig. 2.14-13. The consistent nodal moments at each node are shown as **M**.

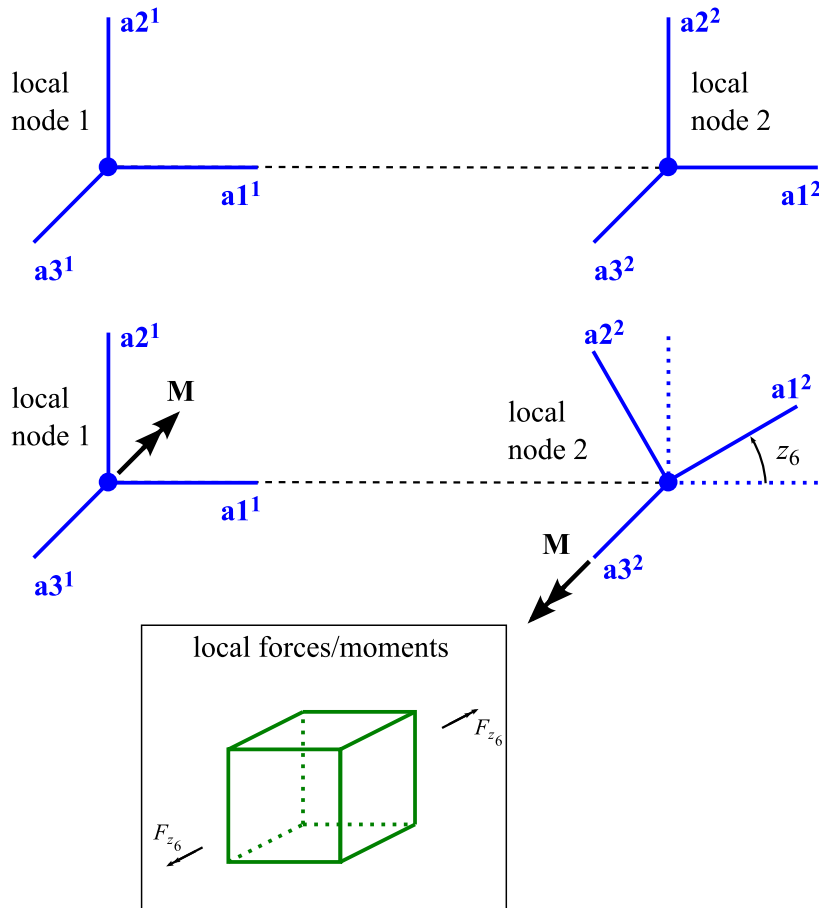


Figure 2.14-13: Illustrative example 2. Joint type connector element with REFERENCE=AXES1 and stiffness associated with rotational degrees of freedom. As node 2 rotates, the element generates a local moment F_{z_6} , as shown. The consistent nodal moments are shown as \mathbf{M} .

Illustrative example 4

Joint type connector element with local node 1 undergoing a rotation with local node 2 fixed in the global system: As a converse of Illustrative example 3, consider now the case of a connector element with node 1 undergoing a rotation relative to node 2, and with no translations with respect to the global system. The reference axes are defined as the a-triad at node 1 (REFERENCE=AXES1). This element will generate a negative internal moment (due to local node 2's negative rotation relative to the local system $\alpha_6 = -\theta$), but it will also generate internal translational forces, as shown in Fig. 2.14-14. This is because as local node 1 rotates, the reference axes rotate with it, and so local node 2's coordinates relative to the local reference system are $x_1 < 0$ and $x_2 < 0$. The consistent nodal forces and moments at each node are shown as \mathbf{F} , \mathbf{M}_1 , and \mathbf{M}_2 , where $\mathbf{M}_1 \neq \mathbf{M}_2$.

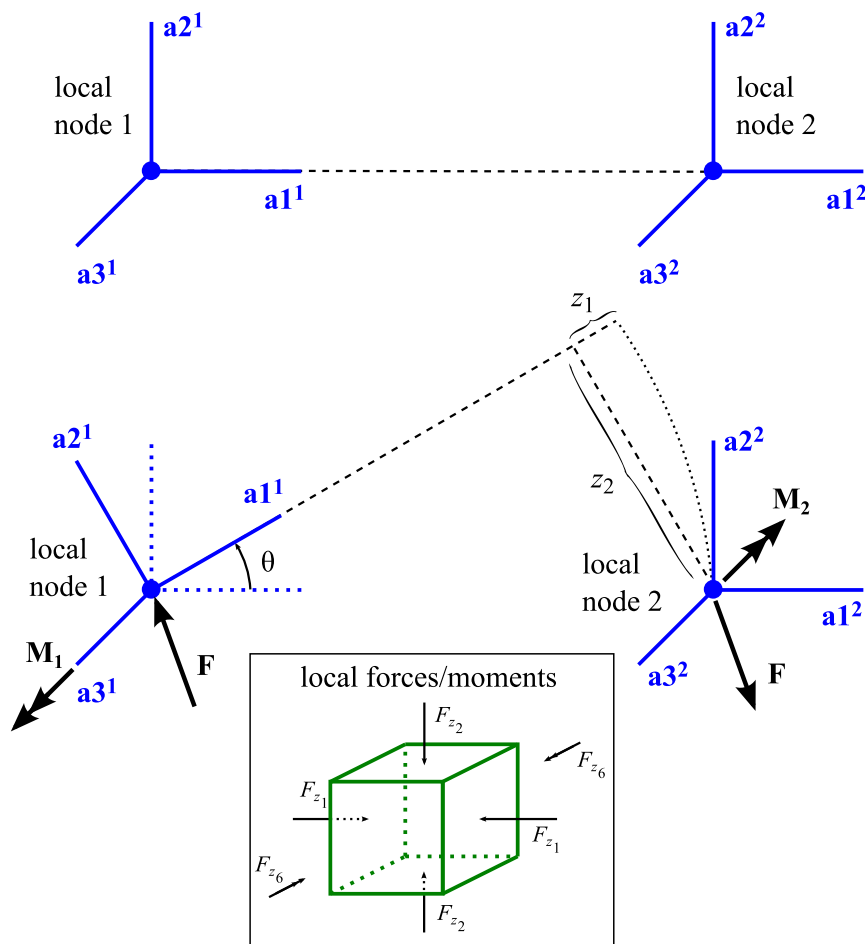


Figure 2.14-14: Illustrative example 4. Joint type connector element with `REFERENCE=AXES1` with local node 1 undergoing a rotation and with local node 2 fixed in the global system. As local node 1 rotates, the reference axes also rotate. Thus, local node 2 rotates and translates with respect to the local reference system. Note also that node 2 undergoes a rotation of $z_6 = -\theta$ with respect to the local reference system.

Comparing Illustrative examples 3 and 4 demonstrates an inherent asymmetry of `REFERENCE=AXES1`. Thus, care must be taken when considering local node numbering when creating connector elements using this option.

Illustrative example 5

Joint type connector element with UNDEFORMED=LOCALORIGIN with nodes undergoing relative translations and rotations: Recall that the option UNDEFORMED=LOCALORIGIN sets all undeformed local displacements to zero. If a connector element with this option is defined with its nodes initially separated by some distance, and if the element's local properties resist translational motions, then the element will generate internal forces tending to bring the local nodes together. Furthermore, if the element's local properties also resist rotational motions, then the element will generate internal moments tending to align the local a-triads. Fig. 2.14-15 shows that as local node 1 rotates, the reference axes rotate with it, and so local node 2's coordinates relative to the local reference system are z_1 , z_2 , and $z_6 = -\theta$ giving rise to corresponding conjugate local forces. In fact, any relative motions between nodes 1 and 2 will generate local forces. The consistent nodal forces and moments at each node are shown as \mathbf{F} , and \mathbf{M} , respectively.

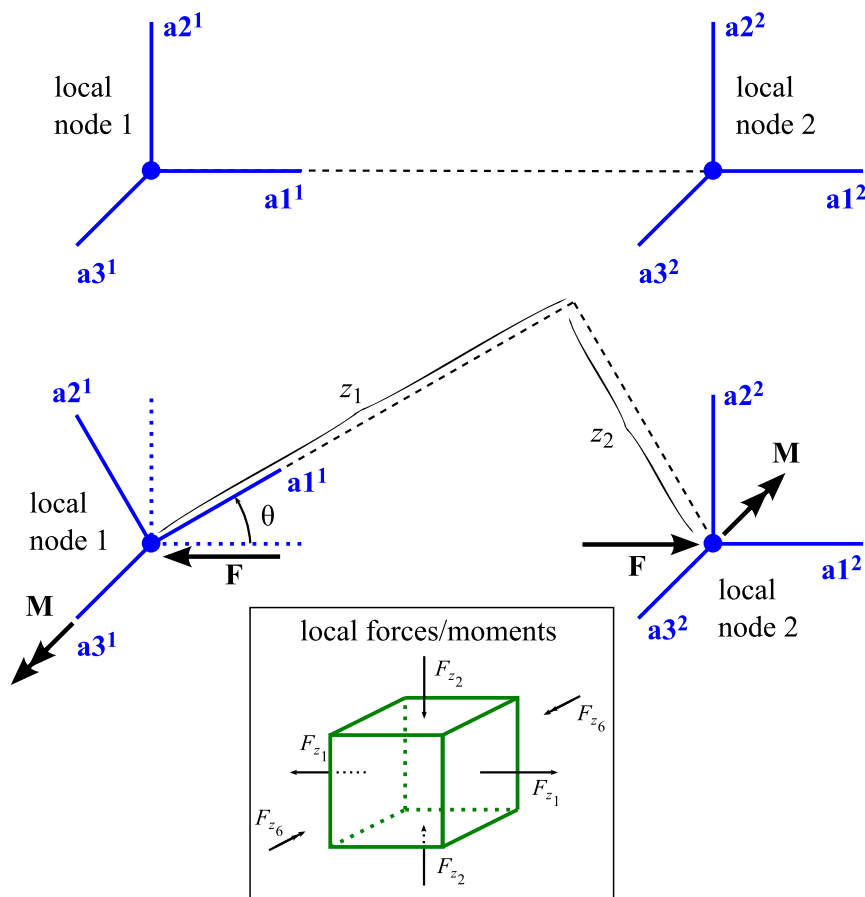


Figure 2.14-15: Illustrative example 5. Joint type connector element with REFERENCE=AXES1 and UNDEFORMED=LOCALORIGIN with local node 1 undergoing a rotation and with local node 2 fixed in the global system. As local node 1 rotates, the reference axes also rotate. Due to the selected undeformed configuration, local node 2 has motions $z_1 > 0$, and $z_2 < 0$ and rotates by $z_6 = -\theta$ with respect to the local reference system.

Illustrative example 6

Joint type connector element with UNDEFORMED=LOCALORIGIN with nodes undergoing relative translations and rotations but no resistance to rotations about local 3-direction: This example is similar to Illustrative example 5. However, the local rotational stiffness about the 3-direction is always zero. Thus, the element does not generate a local moment for any z_6 . Fig. 2.14-16 shows that as local node 1 rotates, the reference axes rotate with it, and so local node 2 undergoes translational displacements z_1 and z_2 relative to the local reference system, giving rise to corresponding conjugate local translational forces. Although node 2 rotates by $z_6 = -\theta$ relative to node 1, no local moment is generated. If the element has stiffnesses associated with the other rotational degrees of freedom, then the element will generate local moments to maintain the nodal 3-directions aligned (but permit rotation about the local 3-direction). The consistent nodal forces and at each node are shown as \mathbf{F} . Note that for this element, no consistent nodal point moments are generated when $f_{z_4}(0) = f_{z_5}(0) = 0$.

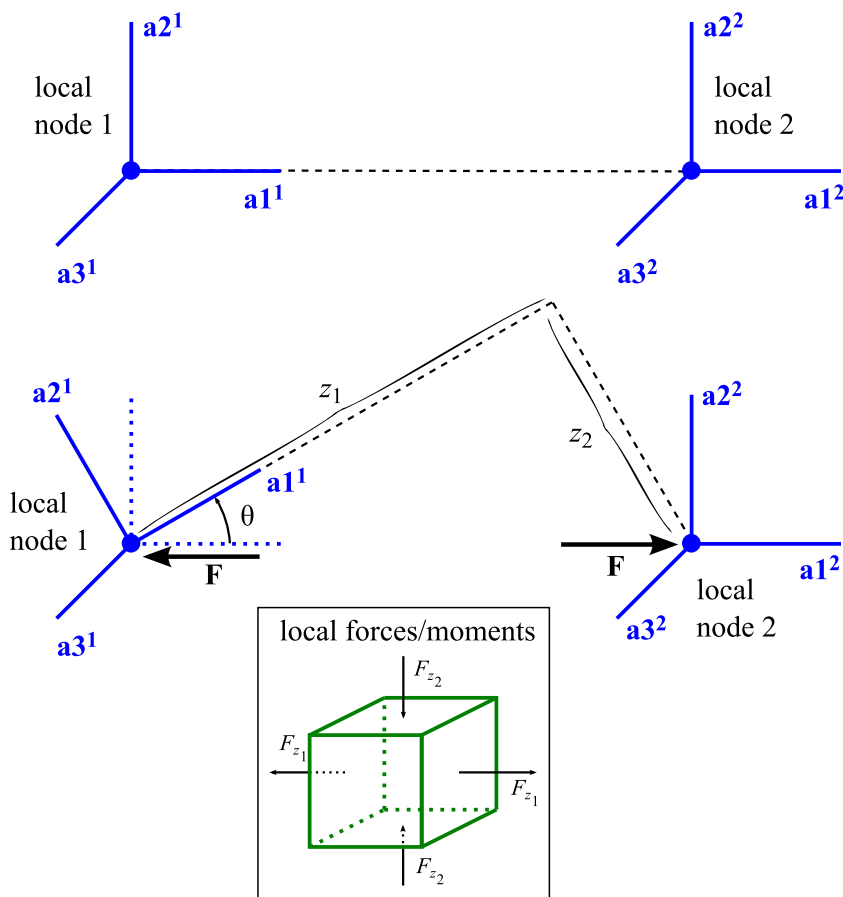


Figure 2.14-16: Illustrative example 6. Joint type connector element with REFERENCE=AXES1 and UNDEFORMED=LOCALORIGIN with local node 1 undergoing a rotation and with local node 2 fixed in the global system. As local node 1 rotates, the reference axes also rotate. Due to the selected undeformed configuration, local node 2 has motions $z_1 > 0$, and $z_2 < 0$ and rotates by $z_6 = -\theta$ with respect to the local reference system. This element does not resist rotations about the local 3-direction, and thus no local moment (or consistent nodal moments) is generated.

Illustrative example 7

Joint type connector element with UNDEFORMED=B1PROJECT with local node 2 undergoing relative translations: Consider the element shown in Fig. 2.14-17 with UNDEFORMED=B1PROJECT. The element undergoes no local rotations and is assumed to have nonzero stiffnesses associated with all local translations such that $f_{z_1}(0) = f_{z_2}(0) = f_{z_3}(0) = 0$. Recall that with this undeformed option, the element is undeformed when $l_1 = L_1^0$ and l_2 and l_3 are zero (see Fig. 2.14-7). Thus, the element shown in Fig. 2.14-17 undergoes local displacements z_2 and z_3 , generating corresponding local conjugate forces. The consistent nodal forces and at each node are shown as **F** and the consistent nodal moment is **M**.

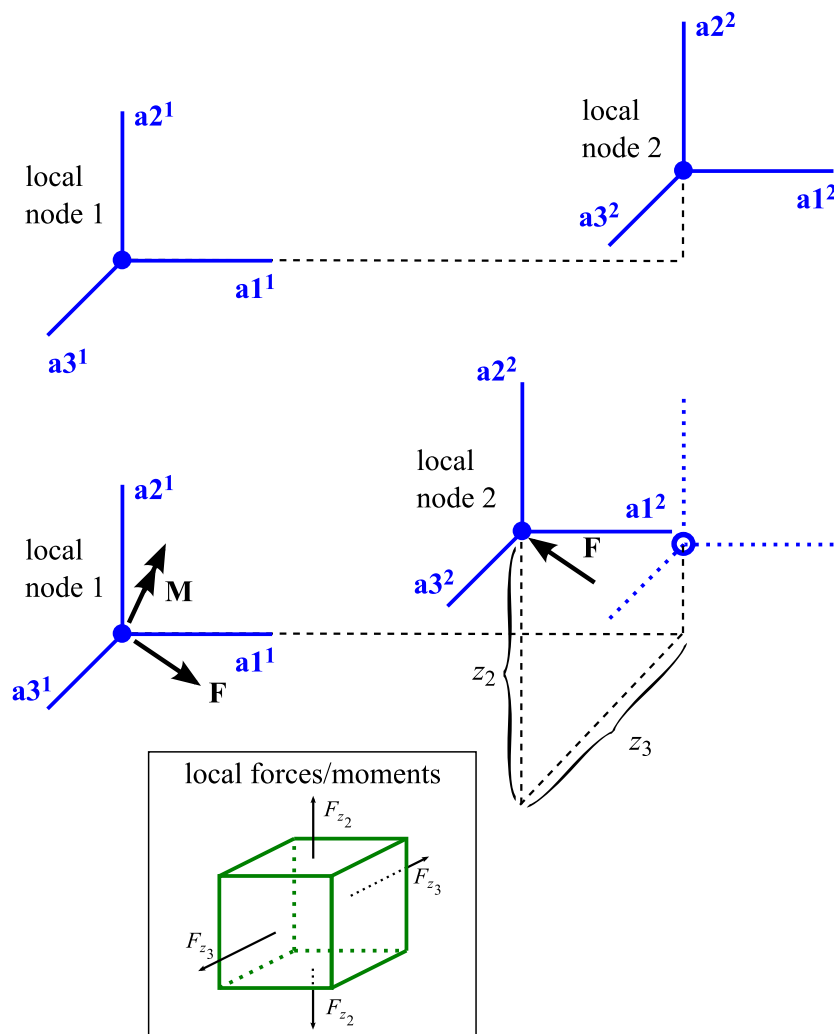


Figure 2.14-17: Illustrative example 7. Joint type connector element with `REFERENCE=AXES1` and `UNDEFORMED=B1PROJECT` with local node 2 undergoing relative translations in the local 2 and 3 directions. Note that the translation z_2 is measured with respect to the projection of node 2's original location onto the $b1$ axis, and likewise for z_3 .

Illustrative example 8

Connector element with `REFERENCE=AXESAVG` with node 1 undergoing a rotation and node 2 fixed in the global system:
Consider the case of a connector element with `REFERENCE=AXESAVG`. As with Illustrative example 4, this element will generate a negative internal moment (due to local node 2's negative rotation relative to the local system $z_6 = -\theta$). It will also generate internal translational forces, as shown in Fig. 2.14-18. Note, however, that in contrast to Illustrative example 4, the local displacements are calculated with respect to the *averaged* b-triad (shown in green), which is not oriented in the same direction as the a-triad at node 1. The consistent nodal forces and moments at each node are shown as \mathbf{F} , \mathbf{M}_1 , and \mathbf{M}_2 , where $\mathbf{M}_1 \neq \mathbf{M}_2$.

The asymmetry demonstrated by comparing Illustrative examples 3 and 4 (with `REFERENCE=AXES1`) is not present when `REFERENCE=AXESAVG`. This is because when `REFERENCE=AXES1`, the orientation of the reference axes depends only upon the orientation of local node 1 and *not* local node 2, but when `REFERENCE=AXESAVG`, the orientation of the reference axes depends upon the orientations of local nodes 1 *and* 2.

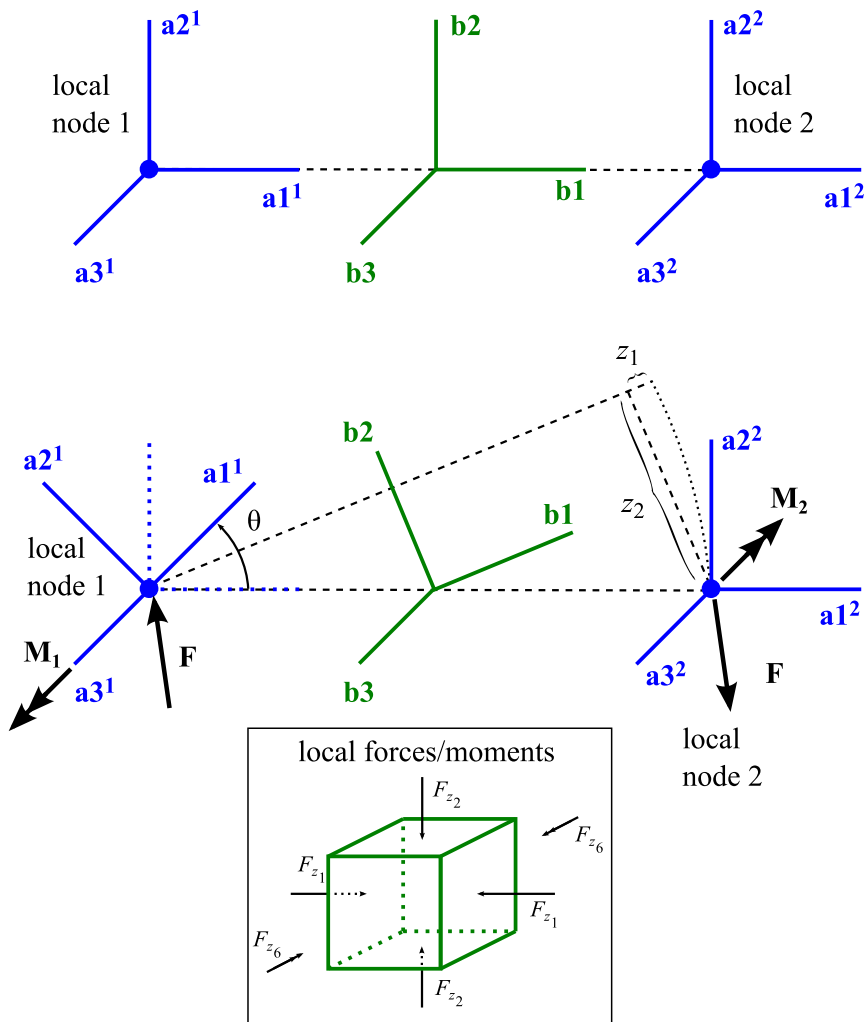


Figure 2.14-18: Illustrative example 8. Connector element with REFERENCE=AXESAVG. In this case, the moment generated is proportional to the angle θ . The distance between the nodes has increased, and the reference axes have rotated. Thus, translations forces develop in the element's 1- and 2-directions, as shown.

It may surprise the reader to see that the consistent nodal moments $\mathbf{M}_1 \neq \mathbf{M}_2$. Recalling the principle of virtual work for planar motions

$$F_{z_6} \delta z_6 = F_1^1 \delta u_1^1 + F_2^1 \delta u_2^1 + M_3^1 \delta \theta_3^1 + F_1^2 \delta u_1^2 + F_2^2 \delta u_2^2 + M_3^2 \delta \theta_3^2 .$$

By setting $\delta \theta_3^1 \neq 0$ only, we obtain the equation $F_{z_6} \delta z_6 = M_3^1 \delta \theta_3^1$, and Fig. 2.14-19 conceptualizes how $\delta \theta_3^1$ affects δz_6 in the final configuration.

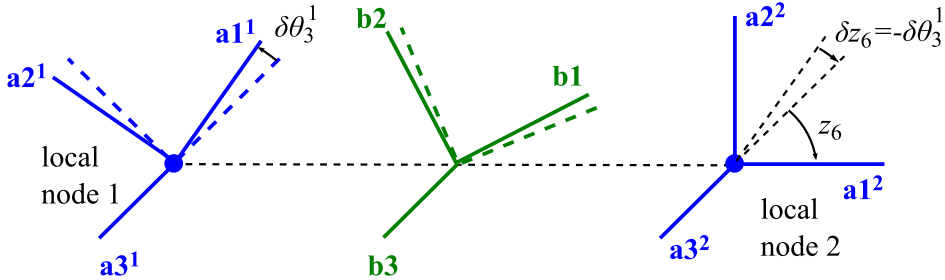


Figure 2.14-19: Conceptualization of how variations in θ_3^1 affect variations in z_6 about the connector element's final configuration. In this case, it is clear that $\delta z_6 = -\delta \theta_3^1$.

Setting $\delta \theta_3^2 \neq 0$ only, we now obtain the equation

$$F_{z_6} \delta z_6 = M_3^2 \delta \theta_3^2 . \text{ See Fig. 2.14-20.}$$

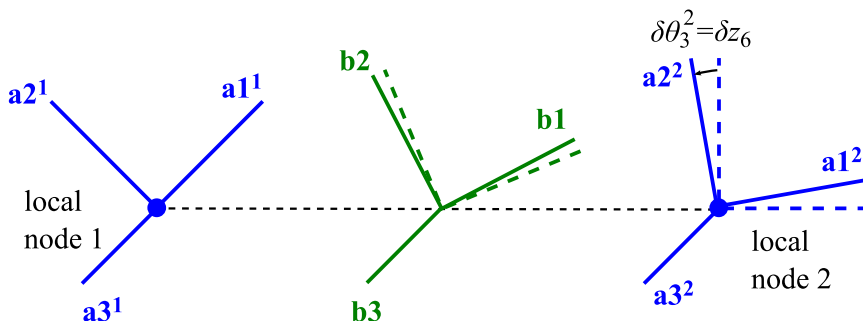


Figure 2.14-20: Conceptualization of how variations in θ_3^2 affect variations in z_6 about the connector element's final configuration. In this case, it is clear that $\delta z_6 = \delta \theta_3^2$.

By comparing Figs. 2.14-19 and 2.14-20, it can be seen that variations in θ_3^1 will have a different effect on variations in z_6 than will θ_3^2 . Thus, we should expect different consistent nodal moments at local nodes 1 and 2.

2.14.3 Modeling

2.14.3.1 Specification of element groups properties

The `EGROUP CONNECTOR` command is used to define the basic characteristics of a connector element group. For example, this command specifies the element subtype (`JOINT`, `MATRIX`, or `MATRIX-MULTILINEAR`), printing and saving options, the default time of element death, the connector element property set number, and the time function label used for scaling the element group's properties. `EGROUP CONNECTOR` also specifies the reference axis system options (`AXES1` or `AXESAVG`).

The reference axes are specified at the element group level using `EGROUP CONNECTOR REFERENCE`. The default option for `JOINT` elements is to have the reference axes translate and rotate with local

node 1 (AXES1). Because the reference axes system (i.e., the element's local datum) can be specified to move with local node 1 (but not local node 2), this option will result in the connector element behaving asymmetrically. Thus, two elements differing only in the order of the local nodes will in general give different results, as can be seen by comparing Illustrative examples 3 and 4 above.

The default option for MATRIX and MATRIX-MULTILINEAR elements is for the reference axes to be the average (AXESAVG) of the a-triads at local nodes 1 and 2.

Connector element properties can be scaled as a function of time using EGROUP TIMEFUNCTION. Thus, the connector element can generate internal forces to resemble a prescribed load or constraint.

2.14.3.2 Specification of joint type properties

The CONN-PROP JOINT command allows the user to specify if the joint type connector element is to be of TYPE option LINEAR or NONLINEAR.

When defining a materially linear joint property set using CONN-PROP JOINT TYPE=LINEAR, the individual joint stiffness and damping coefficients for each degree of freedom are input directly using K1, ..., K6 and C1, ..., C6, respectively.

When defining a materially nonlinear joint property set using CONN-PROP JOINT TYPE=NONLINEAR, labels for the individual multilinear joint stiffness and damping responses for each degree of freedom are specified using KNL1, ..., KNL6 and CNL1, ..., CNL6, respectively.

In both cases, the numerals correspond to local degrees of freedom, as follow

DOF 1 = relative displacement z_1 (or velocity \dot{z}_1) in local 1 direction

DOF 2 = relative displacement z_2 (or velocity \dot{z}_2) in local 2

direction

DOF 3 = relative displacement z_3 (or velocity \dot{z}_3) in local 3

direction

DOF 4 = relative rotation z_4 (or angular velocity \dot{z}_4) about local

1 direction

DOF 5 = relative rotation z_5 (or angular velocity \dot{z}_5) about local

2 direction

DOF 6 = relative rotation z_6 (or angular velocity \dot{z}_6) about local

3 direction

Properties relating to rotational stiffness or damping behavior are defined with respect to `UNITANGLE`, which may either be `RADIANS` or `DEGREES`.

Multilinear approximations to nonlinear behavior are defined using the commands `JOINT-NL-K` (for defining forces or moments vs displacements or rotations) and `JOINT-NL-C` (forces or moments vs translational or angular velocities).

These commands have identical syntaxes, so it suffices to discuss one command. The following examples show force-displacement definitions, but the same concepts apply for moment-rotation, force-velocity, and moment-velocity definitions. When specifying `KNL4-KNL6` and/or `CNL4-CNL6` in `CONN-PROP JOINT`, the angles or angular velocities are interpreted consistently with `UNITANGLE`. The program linearly interpolates the response between user-defined data points.

When the `JOINT-NL-K SYMMETRY=YES` option is used, the user must specify only the force-displacement behavior for positive displacements. The program will automatically construct a symmetric multilinear relation for negative displacements. In this case, the curves must pass through the origin. See Fig. 2.14-21.

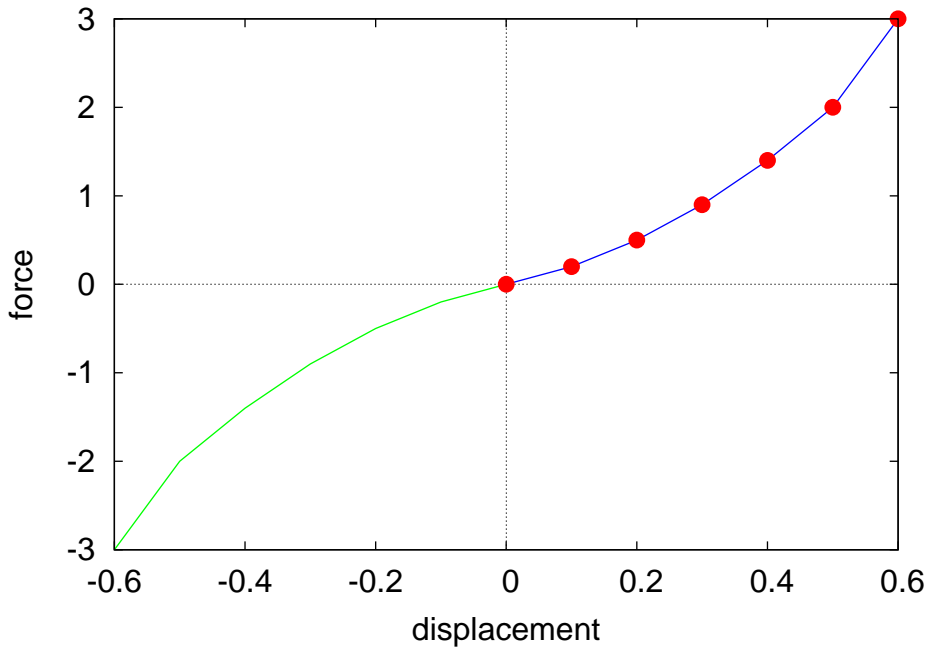


Figure 2.14-21: Illustration of a multilinear force-displacement definition as interpreted by the program when `SYMMETRY=YES`. User-supplied input for non-negative displacements is shown by red circles, and the response between each point is linearly interpolated (linear blue segments). The symmetric response for negative displacements (which is automatically assumed by the program when `SYMMETRY=YES`) is denoted by the linear green segments.

Setting `SYMMETRY=NO` allows the user to specify nonsymmetric behavior, which need not pass through the origin. See Fig. 2.14-22. By defining responses which do not pass through their origins, it is possible to create connector elements with arbitrary undeformed states for each degree of freedom separately.

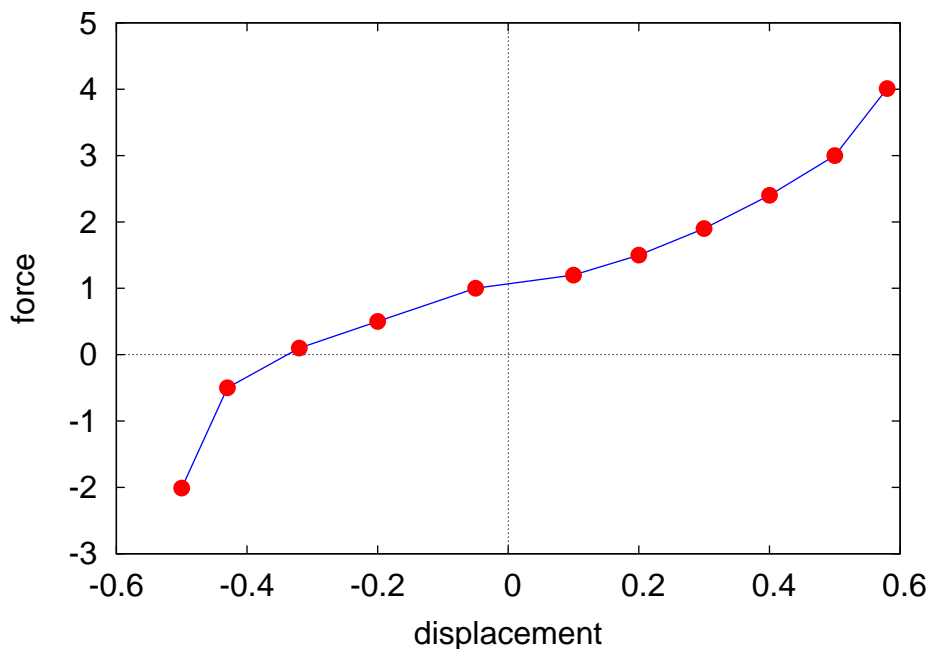


Figure 2.14-22: Illustration of a multilinear force-displacement definition entered with SYMMETRY=NO. Note that the defined response need not pass through the origin. This “offset” can be used to effectively define arbitrary undeformed configurations for each separate degree of freedom.

The user can specify the desired behavior when the element undergoes motions beyond its defined force-displacement definitions using the LIMITS option. By default, LIMITS=EXTRAPOLATE will simply extrapolate the stiffness defined by the 2 data points defined by the user nearest the limit. See Fig. 2.14-23.

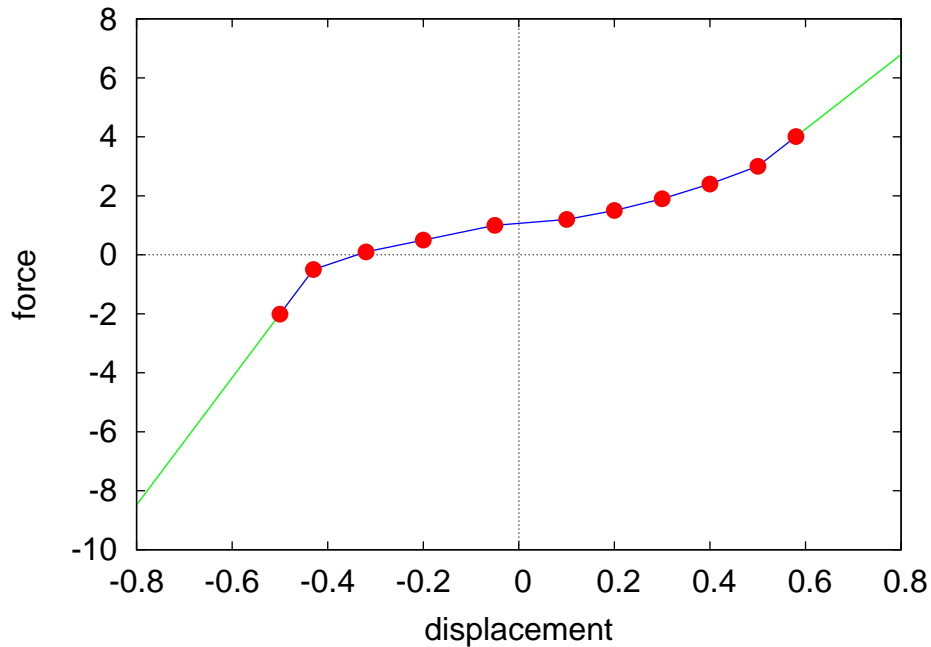


Figure 2.14-23: Illustration of how user-defined data (red circles) is extrapolated when `LIMITS=EXTRAPOLATE`. The response beyond the user-defined limits (green lines) is extrapolated by the stiffness defined by the 2 data points nearest each limit. As illustrated by this nonsymmetric example, the extrapolated responses beyond the limits need not be of equal slope.

Setting `LIMITS=BACKSTOP` will result in the stiffness behavior (that is, the slope of the force-displacement curve) beyond the user-defined limits to be set to the damping coefficient at the defined limit scaled by `BACKFAC`. See Fig. 2.14-24.

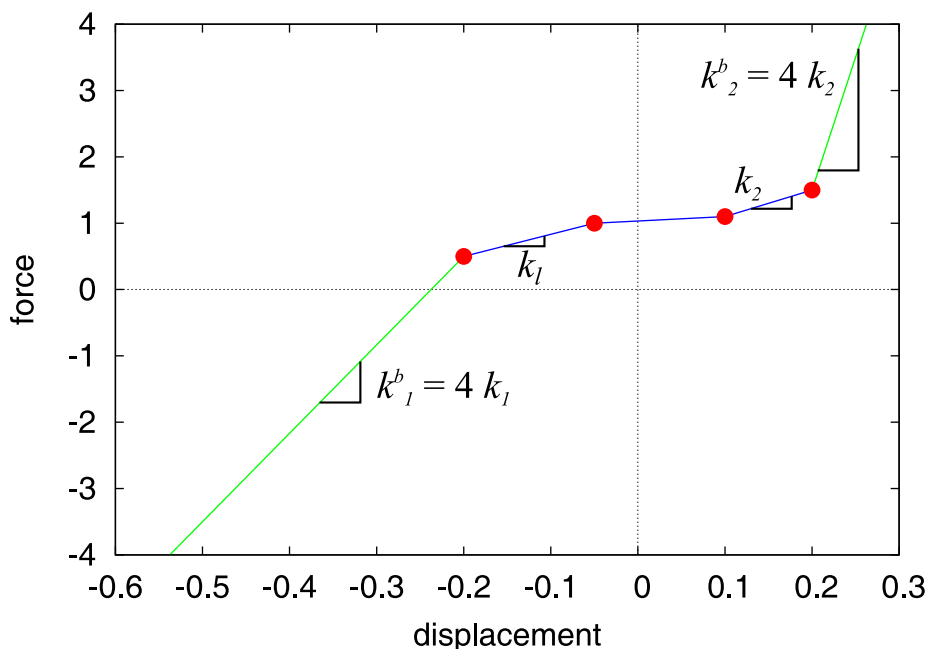


Figure 2.14-24: Illustration of how the program computes force-displacement response when `LIMITS=BACKSTOP` and `BACKFAC=4.0` (relatively small factor selected for ease of viewing). The stiffness k_1 (at the leftmost end of the defined response) is scaled by `BACKFAC` and extended beyond the leftmost data point. Similarly, the stiffness k_s (at the rightmost end of the defined response) is scaled by `BACKFAC` and extended beyond the rightmost data point. In practice, large values of `BACKFAC` can be used to effectively model a firm backstop.

Note that, by default, `BACKFAC` is `1.0E10`, which may be too large for some problems and prevent convergence when the displacement exceeds the defined limits. If this occurs, define `BACKFAC` such that the backstop effect is sufficiently stiff for the purposes of the model but not excessively stiff that it prevents convergence.

2.14.3.3 Specification of matrix type properties

The `CONN-PROP MATRIX` command allows the user to specify

properties for MATRIX connector elements. The user must specify the type(s) of matrices being input and if the matrices are symmetric or nonsymmetric.

Properties relating to rotational stiffness or damping behavior are defined with respect to radians only.

CONN-PROP MATRIX TYPE specifies if the STIFFNESS matrix, the DAMPING matrix, or if BOTH matrices are being entered. Matrix symmetry is specified using CONN-PROP MATRIX SYMMETRIC={YES/NO}. The following is a demonstration of how a symmetric stiffness matrix can be input:

```
CONN-PROP MATRIX NAME=1 TYPE=STIFFNESS
1  1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9  1.10  1.11  1.12
2  0.0 2.2 2.3 2.4 2.5 2.6 2.7 2.8 2.9  2.10  2.11  2.12
3  0.0 0.0 3.3 3.4 3.5 3.6 3.7 3.8 3.9  3.10  3.11  3.12
4  0.0 0.0 0.0 4.4 4.5 4.6 4.7 4.8 4.9  4.10  4.11  4.12
5  0.0 0.0 0.0 0.0 5.5 5.6 5.7 5.8 5.9  5.10  5.11  5.12
6  0.0 0.0 0.0 0.0 0.0 6.6 6.7 6.8 6.9  6.10  6.11  6.12
7  0.0 0.0 0.0 0.0 0.0 0.0 7.7 7.8 7.9  7.10  7.11  7.12
8  0.0 0.0 0.0 0.0 0.0 0.0 0.0 8.8 8.9  8.10  8.11  8.12
9  0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 9.9  9.10  9.11  9.12
10 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 10.10 10.11 10.12
11 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  0.0 11.11 11.12
12 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  0.0  0.0 12.12
```

The floating point entries are in row.column format for clarity. Note that by default, CONN-PROP MATRIX SYMMETRIC=YES, and so in this case, the lower triangular portion of the matrix (entered as zeroes) is ignored. Also note the first column of integers 1 . . 12 designating the i^{th} input row. For cases where TYPE=BOTH, this first column will run from 1 . . 24 (12 rows each for the stiffness and damping matrices). A matrix property set so defined can then be referenced using:

```
EGROUP CONNECTOR SUBTYPE=MATRIX CONN-PROP=1
```

The degrees of freedom for the user-defined 12x12 matrices are:

1-3 = displacements of local node 1
4-6 = rotations of local node 1
7-9 = displacements of local node 2
10-12 = rotations of local node 2

So, for example, the lower right 6x6 submatrix can be interpreted as defining the response of local node 2 when node 1 is fixed.

2.14.3.4 Specification of multilinear matrix type properties

The `CONN-PROP MATRIX-MULTILINEAR` command allows the user to specify properties for `MATRIX-MULTILINEAR` connector elements. The user must specify the nonlinear stiffness and/or damping matrix definitions being used via the `CONN-PROP MULTI-MATRIX KNL` and `CNL` parameters.

The multilinear behaviors are defined using the `MATRIX-NL-K` and `MATRIX-NL-C` commands. Each nonlinear stiffness (or damping) definition includes parameters specifying the `TYPE` of nonlinear response (that is, if the response depends upon the relative displacements, velocities, local elastic or damping force, or temperature), if the nonlinear behavior is or is not `SYMMETRIC`, and the behavior beyond the `LIMITS` of the multilinear response (extrapolated or if the last matrix is extended).

The stiffness and damping matrices to be referenced are defined using commands `MATRIX-K` and `MATRIX-C`, respectively. Note that these commands have their own `SYMMETRIC` parameters, which specify if the matrix is or is not symmetric. This is not the same as the `SYMMETRIC` parameters used in the `MATRIX-NL-K` and `MATRIX-NL-C` commands which specify the symmetry of the multilinear response. Each command takes as input either the full 12x12 matrix (when `SYMMETRIC=NO`) or the upper triangular portion when `SYMMETRIC=YES`. When `SYMMETRIC=YES`, the lower triangular portion of the matrix is ignored. The following input defines a symmetric damping matrix for use in a multilinear matrix connector element:

```

MATRIX-C NAME=1
1  1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9  1.10  1.11  1.12
2  0.0 2.2 2.3 2.4 2.5 2.6 2.7 2.8 2.9  2.10  2.11  2.12
3  0.0 0.0 3.3 3.4 3.5 3.6 3.7 3.8 3.9  3.10  3.11  3.12
4  0.0 0.0 0.0 4.4 4.5 4.6 4.7 4.8 4.9  4.10  4.11  4.12
5  0.0 0.0 0.0 0.0 5.5 5.6 5.7 5.8 5.9  5.10  5.11  5.12
6  0.0 0.0 0.0 0.0 0.0 6.6 6.7 6.8 6.9  6.10  6.11  6.12
7  0.0 0.0 0.0 0.0 0.0 0.0 7.7 7.8 7.9  7.10  7.11  7.12
8  0.0 0.0 0.0 0.0 0.0 0.0 0.0 8.8 8.9  8.10  8.11  8.12
9  0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 9.9  9.10  9.11  9.12
10 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 10.10 10.11 10.12
11 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  0.0 11.11 11.12
12 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  0.0  0.0 12.12

```

The floating point entries are in `row.column` format for clarity. Note that by default, `MATRIX-K` and `MATRIX-C SYMMETRIC=YES`, and so in this case, the lower triangular portion of the damping matrix (entered as zeroes) is ignored. Also note the first column of integers 1 . . 12 designating the i^{th} input row.

Once the stiffness and/or damping matrices have been input, it is a simple matter to define the multilinear behavior using `MATRIX-NL-K` and `MATRIX-NL-C`. For example, the following nonlinear stiffness matrix definition is a function of the element's local 1-displacement and specifies a symmetric response which is extrapolated beyond the defined limit of 0.25. The command references 3 stiffness matrices defined by `MATRIX-K NAME=1`, `MATRIX-K NAME=2`, and `MATRIX-K NAME=3`.

```

MATRIX-NL-K 1 TYPE=DISP-1 SYMMETRIC=YES
LIMITS=EXTRAPOLATE
0.00 1
0.12 2
0.25 3

```

This multilinear stiffness matrix response can be referenced by a connector element property set using

```
CONN-PROP MATRIX-MULTILINEAR KNL=1
```

Finally, the multilinear matrix connector element property set so

defined can then be referenced using

```
EGROUP CONNECTOR SUBTYPE=MATRIX-MULTILINEAR CONN-  
PROP=1
```

2.14.3.5 Specification of element local node triads

As outlined above, the connector element's triad definition consists of three parts, each of which must be specified by the user:

Specification of the a-triad at local node 1

The user can select one of several TRIAD1OPT options in the CONN-DEFINE command. The choices are:

GLOBAL/AXES/AUXPT/AUXNODE/ORIENTATION/AXIAL

GLOBAL: The a-triad at local node 1 is parallel to the global coordinate axes. That is, the **a1**, **a2** and **a3** directions at local node 1 correspond to the global X-, Y-, and Z- directions, respectively. Fig. 2.14-1 shows the case of TRIAD1OPT=GLOBAL.

AXES: The a-triad at local node 1 is defined directly using axes defined by the AXES command. When TRIAD1OPT=AXES, the label defining the axes is specified using the AXES1 option.

AUXPT: The **a1** direction at node 1 is parallel to the vector from local node 1 to local node 2, and the **a2** direction lies in the plane defined by the two nodes and the geometry point label defined by AUXPT1. Direction **a3** is orthogonal.

AUXNODE: The **a1** direction at node 1 is parallel to the vector from local node 1 to local node 2, and the **a2** direction lies in the plane defined by the two nodes and the node label defined by AUXNODE1. Direction **a3** is orthogonal.

ORIENTATION: The **a1** direction at node 1 is parallel to the vector from local node 1 to local node 2, and the **a2** direction lies in the plane defined by the two nodes and an orientation

vector. Direction **a3** is orthogonal. $XO1$, $YO1$, and $ZO1$ must also be specified to define the components of the orientation vector with respect to the coordinate system defined by **O1SYSTEM**. The resulting orientation vector is parallel to that vector which originates at **O1SYSTEM**'s origin and terminates at $(XO1, YO1, ZO1)$. See Fig. 2.14-25.

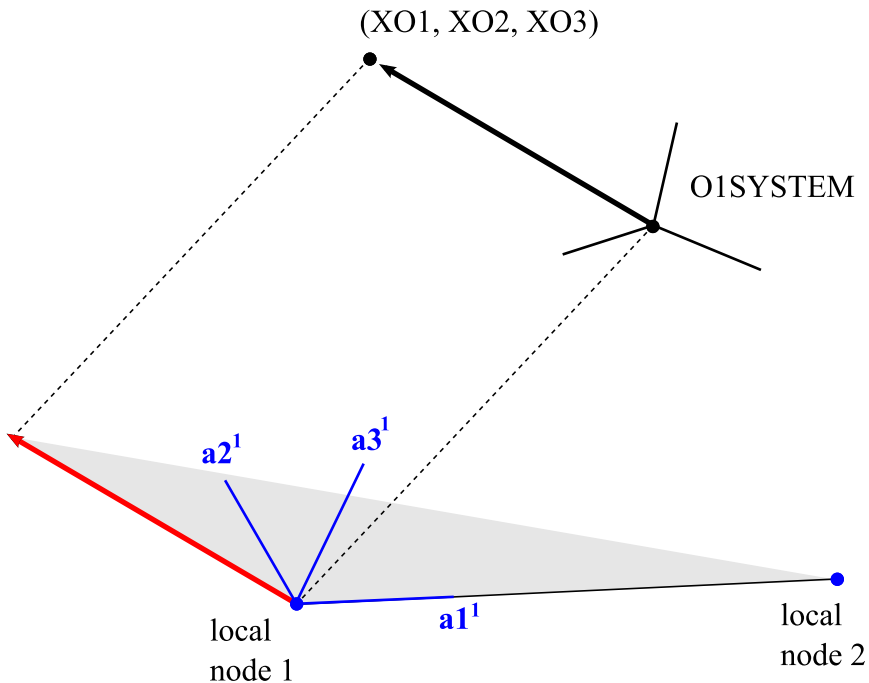


Figure 2.14-25: When using **TRIAD1OPT=ORIENTATION**, the **a1** direction at node 1 is oriented in the direction from node 1 to node 2, but the **a2** direction is defined using an orientation vector. The orientation vector (shown in red) is parallel to the vector defined by coordinates $(XO1, XO2, XO3)$, which is defined in the system **O1SYSTEM**. Direction **a2** lies in the plane defined by local nodes 1 and 2 and the orientation vector (plane shown in gray). The **a3** direction is orthogonal.

AXIAL: The **a1** direction at node 1 is parallel to the vector from local node 1 to local node 2, and the **a2** direction is arbitrary. Direction **a3** is orthogonal. See Fig. 2.14-26.

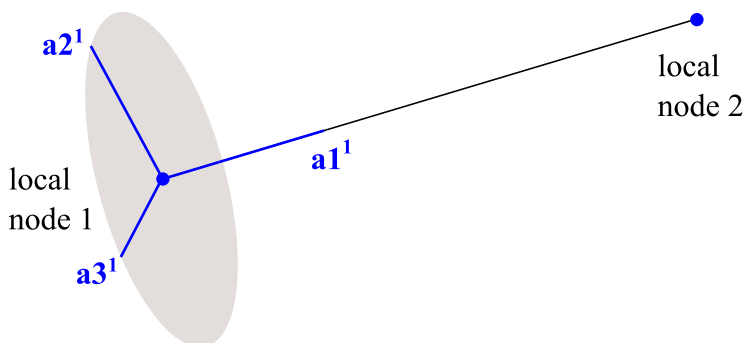


Figure 2.14-26: When using TRIAD1OPT=AXIAL, the **a1** direction at node 1 is oriented in the direction from node 1 to node 2, but the **a2** direction is arbitrary. The **a3** direction is orthogonal. Thus, the **a2** and **a3** directions lie in the plane normal to direction **a1** (depicted in gray).

Specification of the a-triad at local node 2

The a-triad at local node 2 can be specified using TRIAD2OPT in the same ways as for node 1, but an additional option is available: setting TRIAD2OPT=TRIAD1 (the default) uses the same directions of the a-triad defined at local node 1 and places its origin at local node 2.

Specification of the b-triad

The definition for the b-triad (the element's reference system) is specified at the element group level using the EGROUP CONNECTOR REFERENCE option. The choices are:

AXES1: The element's reference system coincides with the a-triad at node 1. This reference system will move with node 1 as it translates and rotates in space. This is the default option for joint type connector elements.

AXESAVG: The orientations of the element's reference system are averaged from the orientations of the a-triads at local nodes 1 and 2. The b-triad is located midway between the connector element's nodes. The reference system moves and rotates as the element's nodes translate and rotate in space. Fig. 2.14-2 shows how this averaging occurs in 2D. This is the default option for matrix type connector elements.

2.14.3.6 Specification of connector element sets

Connector element sets are generated using the `CONN-DEFINE` command. A single element or a "spider" of multiple connector elements can be generated within a connector element set in a variety of ways. For example, connector element sets can be defined using existing points, nodes, nodesets, lines, faces, bodies, etc. For more information about generating connector element spiders, see the Command Reference Manual entry for `CONN-DEFINE`.

Triad definitions at local nodes 1 and 2 are also set using `CONN-DEFINE` using `TRIAD1OPT` and `TRIAD2OPT`, respectively, along with the associated options. See section 2.14.3.5 for detailed information about the various triad options. Note, however, that the element's reference axes are specified at the element group level using `EGROUP CONNECTOR REFERENCE`.

2.14.3.7 Connector element output

The connector element can output consistent nodal point forces and/or local relative displacements (and velocities, when damping behavior is defined in dynamic problems) and local conjugate forces. The selection is made in the `EGROUP CONNECTOR RESULTS = {LOCAL/FORCE/ALL}` parameter.

Printing of local connector element results to the `.out` file requires that `PRINTOUT OUTPUT = ALL`.

Local results output

When a connector element outputs local data, the conjugate quantities are individually output for each degree of freedom relative to the element's current (convecting) reference axes.

In static problems or dynamic problems for which the element has no damping behavior defined, only the relative displacements and rotations and their conjugate elastic forces are output.

In dynamic problems for which the element has damping behavior defined, the relative displacements and velocities are output, along with the corresponding conjugate elastic and damping forces.

The postprocessing variables for the local element results are

```
CONNECTOR_REL_DISPLACEMENT-{123456}  
CONNECTOR_ELASTIC_FORCE-{123456}  
CONNECTOR_REL_VELOCITY-{123456}  
CONNECTOR_DAMPING_FORCE-{123456}  
CONNECTOR_TOTAL_FORCE-{123456}
```

Where the numerals correspond to local degrees of freedom (DOF), as follow:

DOF 1 = relative displacement z_1 (or velocity \dot{z}_1) in local 1 direction

DOF 2 = relative displacement z_2 (or velocity \dot{z}_2) in local 2 direction

DOF 3 = relative displacement z_3 (or velocity \dot{z}_3) in local 3 direction

DOF 4 = relative rotation z_4 (or angular velocity \dot{z}_4) about local 1 direction

DOF 5 = relative rotation z_5 (or angular velocity \dot{z}_5) about local 2 direction

DOF 6 = relative rotation z_6 (or angular velocity \dot{z}_6) about local 3 direction

If a joint type element has DOFs with no corresponding user-defined property sets, then no local element results are saved or printed for those DOFs.

For matrix elements, if the user defines only a stiffness matrix, then only relative displacements and conjugate elastic forces are saved or printed. In the user defines only a damping matrix, then only relative velocities and conjugate damping forces are saved or printed.

Force results output

When a connector element outputs force data, the consistent nodal point forces are output for each of the element's local nodes. The consistent nodal forces are output in terms of the coordinate system used by the node (i.e., the global system or an attached skew system).

All results output

The connector element can output consistent nodal point forces and the local results by setting `EGROUP CONNECTOR RESULTS=ALL`.

Multilinear matrix elements

Multilinear matrix connector elements output additional information for postprocessing. If a multilinear matrix connector element has an associated elastic response, then the `CONNECTOR_STIFFNESS_KEY_TYPE` and `CONNECTOR_STIFFNESS_KEY_VALUE` variables are available for postprocessing. Similarly, if a multilinear matrix connector element has an associated damping response, then the `CONNECTOR_DAMPING_KEY_TYPE` and `CONNECTOR_DAMPING_KEY_VALUE` variables are available for postprocessing.

CONNECTOR_STIFFNESS_KEY_TYPE and CONNECTOR_DAMPING_KEY_TYPE correspond to the MATRIX-NL-K and MATRIX-NL-C TYPE parameters specified at input.

The possible types are as follow:

TEMPERATURE
 (temperature)
DISP-{TMAG/RMAG}; DISP-{123456}
 (displacements)
VEL-{TMAG/RMAG}; VEL-{123456} (velocities)
ELASTIC-{TMAG/RMAG}; ELASTIC-{123456} (elastic
forces)
DAMP-{TMAG/RMAG}; DAMP-{123456} (damping
forces)

Where the numerals correspond to local degrees of freedom (DOF), as before, and TMAG and RMAG correspond to the translational and rotational magnitudes, respectively.

CONNECTOR_STIFFNESS_KEY_VALUE and CONNECTOR_DAMPING_KEY_VALUE are the numerical values corresponding to CONNECTOR_STIFFNESS_KEY_TYPE and CONNECTOR_DAMPING_KEY_TYPE, respectively.

2.14.3.8 Modeling examples

The following section contains several input examples for single elements and includes a brief description of the resulting element and its behavior under certain conditions. This is meant to provide additional insight into how the connector element functions, in practice.

For simplicity, the examples are purely elastic (do not include damping effects), but they will readily generalize to include damping by setting the appropriate damping properties and time integration methods. The reader can examine the following examples by opening the AUI and generating points 1 and 2 some

distance apart and then pasting the commands into the command window. To view the connector element's triads, enter the following commands after generating the element:

```
ELDEPICTION TRIAD=YES  
MESHPLOTT
```

Modeling example 1: Bushing

A joint type connector element is defined with stiffnesses for each translational and rotational degree of freedom. Local nodes 1 and 2 will be located at points 1 and 2, respectively, though these points may or may not be originally coincident. By default, the triads at each local node will be aligned with the global coordinate system, and the element's reference axes will correspond to the a-triad at local node 1. By using the UNDEFORMED=LOCALORIGIN option, the element is undeformed when its nodes are coincident – see Fig. 2.14-6. Consequently, this element will generate local forces to maintain its local nodes coincident and moments to maintain the a-triads at local nodes 1 and 2 aligned. This can be used to effectively model a flexible bushing joint. For more information, see Illustrative example 5.

```
CONN-PROP JOINT NAME=1 K1=12.0 K2=11.0 K3=11.0,  
                      K4=57.0 K5=59.0 K6=57.0  
*  
EGROUP CONNECTOR NAME=1 SUBTYPE=JOINT CONN-PROP=1  
*  
CONN-DEFINE NAME=1 GROUP=1 SIDE1TYPE=POINT  
SIDE1NAME=1,  
          SIDE2TYPE=POINT UNDEFORMED=LOCALORIGIN  
CLEAR  
2 0  
DATAEND
```

Modeling example 2: Bearing

In this example, a joint type connector element is defined with stiffnesses for each translational degree of freedom, but for only 2 rotational degrees of freedom. Local nodes 1 and 2 will be located at points 1 and 2, respectively, and these points may or may not be

originally coincident. By default, the triads at each local node will be aligned with the global coordinate system, and the element's reference axes will correspond to the a-triad at local node 1. By using the `UNDEFORMED=LOCALORIGIN` option, the element is undeformed when its nodes are coincident – see Fig. 2.14-6. Consequently, this element will generate local forces to maintain its local nodes coincident and moments to maintain the nodal 3-directions aligned but not resist rotations about the local 3-direction, because $K_6=0.0$. As a result, this element can be used to effectively model a frictionless bearing with its axis of rotation aligned in the local 3-direction. For more information, see Illustrative example 6.

```
CONN-PROP JOINT NAME=1 K1=1.0e3 K2=1.0e3
K3=1.0e3,
                                K4=1.0e3 K5=1.0e3 K6=0.0
*
EGROUP CONNECTOR NAME=1 SUBTYPE=JOINT CONN-PROP=1
*
CONN-DEFINE NAME=1 GROUP=1 SIDE1TYPE=POINT
SIDE1NAME=1,
            SIDE2TYPE=POINT UNDEFORMED=LOCALORIGIN
CLEAR
2 0
DATAEND
```

Modeling example 3: Slider

This example demonstrates how to use the undeformed option so that the connector element generates forces that allow node 2 to “slide” freely along the direction parallel to the **b2** direction. The `UNDEFORMED=B1PROJECT` option (see Fig. 2.14-7) will set the undeformed configuration ($L_1 = L_1^0, L_2 = 0, L_3 = 0$), but because $K_2=0.0$, the element will not generate a local force F_{z_2} when $z_2 \neq 0$. For more information, see Illustrative example 7.

```
CONN-PROP JOINT NAME=1 K1=1.0e4 K2=0.0 K3=1.0e4,
                                K4=12.0 K5=11.0 K6=12.0
*
EGROUP CONNECTOR NAME=1 SUBTYPE=JOINT CONN-PROP=1
*
CONN-DEFINE NAME=1 GROUP=1 SIDE1TYPE=POINT
```

```
SIDE1NAME=1,  
    SIDE2TYPE=POINT UNDEFORMED=B1PROJECT  
CLEAR  
2 0  
DATAEND
```

Modeling example 4: Convecting matrix element

Before showing a modeling example using a matrix type connector element, it is worth discussing the procedure for obtaining the required matrices. Below, we see how to obtain the 12x12 matrix required for modeling the stiffness response of a general connecting structure. The process for obtaining a damping matrix is similar but requires that the user prescribe a sequence of nodal velocities rather than nodal displacements.

Note: *For a detailed demonstration of obtaining the stiffness matrix for a structure and constructing an equivalent connector element model, see **Primer Problems 67 and 68**.*

To obtain the 12x12 stiffness matrix for a matrix type connector element, the user must already have a 3D model for the connecting structure of interest. The user must then create nodes at the structure's two connecting points and attach each node to the structure (using a rigid link spider, for example). These nodes will represent the corresponding connector element's local nodes. It is then up to the user to decide how the element's triads are to be oriented and to then define the local 1-, 2-, and 3-directions for the connecting structure.

The stiffness matrix entries can then be obtained by sequentially prescribing small displacements in each degree of freedom (while holding all other degrees of freedom fixed) at each (local) connecting node and outputting the reactions at the connecting nodes. For example, the following sequence can be used:

- 1) Displace local node 1 by a small distance in the local 1-direction holding all other displacements and rotations zero. Node 2 is fixed.
- 2) Displace local node 1 by a small distance in the local 2-direction

- holding all other displacements and rotations zero. Node 2 is fixed.
- 3) Displace local node 1 by a small distance in the local 3-direction holding all other displacements and rotations zero. Node 2 is fixed.
 - 4) Rotate local node 1 by a small angle about the local 1-direction holding all other displacements and rotations zero. Node 2 is fixed.
 - 5) Rotate local node 1 by a small angle about the local 2-direction holding all other displacements and rotations zero. Node 2 is fixed.
 - 6) Rotate local node 1 by a small angle about the local 3-direction holding all other displacements and rotations zero. Node 2 is fixed.

Then repeat steps 1-6 for local node 2 while holding local node 1 fixed.

For each of the 12 prescribed displacements, obtain the $\{1/2/3\}$ -reaction forces and $\{1/2/3\}$ -reaction moments at local nodes 1 and 2. The matrix entries are then obtained by solving the 12 equations associated with each load case.

Recall the matrix equation

$$\begin{bmatrix} F_1^1 \\ F_2^1 \\ F_3^1 \\ M_1^1 \\ M_2^1 \\ M_3^1 \\ F_1^2 \\ F_2^2 \\ F_3^2 \\ M_1^2 \\ M_2^2 \\ M_3^2 \end{bmatrix} = \begin{bmatrix} K_{1,1} & K_{1,2} & K_{1,3} & K_{1,4} & K_{1,5} & K_{1,6} & K_{1,7} & K_{1,8} & K_{1,9} & K_{1,10} & K_{1,11} & K_{1,12} \\ K_{2,1} & K_{2,2} & K_{2,3} & K_{2,4} & K_{2,5} & K_{2,6} & K_{2,7} & K_{2,8} & K_{2,9} & K_{2,10} & K_{2,11} & K_{2,12} \\ K_{3,1} & K_{3,2} & K_{3,3} & K_{3,4} & K_{3,5} & K_{3,6} & K_{3,7} & K_{3,8} & K_{3,9} & K_{3,10} & K_{3,11} & K_{3,12} \\ K_{4,1} & K_{4,2} & K_{4,3} & K_{4,4} & K_{4,5} & K_{4,6} & K_{4,7} & K_{4,8} & K_{4,9} & K_{4,10} & K_{4,11} & K_{4,12} \\ K_{5,1} & K_{5,2} & K_{5,3} & K_{5,4} & K_{5,5} & K_{5,6} & K_{5,7} & K_{5,8} & K_{5,9} & K_{5,10} & K_{5,11} & K_{5,12} \\ K_{6,1} & K_{6,2} & K_{6,3} & K_{6,4} & K_{6,5} & K_{6,6} & K_{6,7} & K_{6,8} & K_{6,9} & K_{6,10} & K_{6,11} & K_{6,12} \\ K_{7,1} & K_{7,2} & K_{7,3} & K_{7,4} & K_{7,5} & K_{7,6} & K_{7,7} & K_{7,8} & K_{7,9} & K_{7,10} & K_{7,11} & K_{7,12} \\ K_{8,1} & K_{8,2} & K_{8,3} & K_{8,4} & K_{8,5} & K_{8,6} & K_{8,7} & K_{8,8} & K_{8,9} & K_{8,10} & K_{8,11} & K_{8,12} \\ K_{9,1} & K_{9,2} & K_{9,3} & K_{9,4} & K_{9,5} & K_{9,6} & K_{9,7} & K_{9,8} & K_{9,9} & K_{9,10} & K_{9,11} & K_{9,12} \\ K_{10,1} & K_{10,2} & K_{10,3} & K_{10,4} & K_{10,5} & K_{10,6} & K_{10,7} & K_{10,8} & K_{10,9} & K_{10,10} & K_{10,11} & K_{10,12} \\ K_{11,1} & K_{11,2} & K_{11,3} & K_{11,4} & K_{11,5} & K_{11,6} & K_{11,7} & K_{11,8} & K_{11,9} & K_{11,10} & K_{11,11} & K_{11,12} \\ K_{12,1} & K_{12,2} & K_{12,3} & K_{12,4} & K_{12,5} & K_{12,6} & K_{12,7} & K_{12,8} & K_{12,9} & K_{12,10} & K_{12,11} & K_{12,12} \end{bmatrix} \begin{bmatrix} u_1^1 \\ u_2^1 \\ u_3^1 \\ \theta_1^1 \\ \theta_2^1 \\ \theta_3^1 \\ u_1^2 \\ u_2^2 \\ u_3^2 \\ \theta_1^2 \\ \theta_2^2 \\ \theta_3^2 \end{bmatrix}$$

,

where F_1^i, F_2^i, F_3^i and M_1^i, M_2^i, M_3^i are the nodal reaction forces and moments, respectively, at local node i , and u_1^i, u_2^i, u_3^i and $\theta_1^i, \theta_2^i, \theta_3^i$ are the sequentially prescribed translations and rotations, respectively, also at local node i .

When $u_1^1 \neq 0$ and

$u_2^1 = u_3^1 = \theta_1^1 = \theta_2^1 = \theta_3^1 = u_2^2 = u_3^2 = \theta_1^2 = \theta_2^2 = \theta_3^2 = 0$, for example, the original set of 12 linear equations, each with 12 unknowns, simplifies to a set of 12 equations, each with a single unknown

$$K_{j,1} = \frac{F_j^1}{u_1^1},$$
$$K_{j+3,1} = \frac{M_j^1}{u_1^1}, \text{ where } j = 1 \dots 3.$$
$$K_{j+6,1} = \frac{F_j^2}{u_1^1},$$
$$K_{j+9,1} = \frac{M_j^2}{u_1^1},$$

Given the 12 reactions (6 at each node) and u_1^1 , we can see that each load case will furnish the entries of a column in the user-defined stiffness matrix.

The following example demonstrates the input for a symmetric matrix type connector element. In this case, the element is being used to model a straight section of elastic pipe, with the **b1** axis oriented along the pipe's axial direction (hence the use of the TRIAD1OPT=AXIAL option). The initial distance between points 1 and 2 corresponds to the pipe's undeformed length. During the analysis, this element may undergo large rotations in space. However, the element's properties will convect with the element. For example, if the element were to undergo rigid body rotations, its local forces would remain unchanged.

```

CONN-PROP MATRIX NAME=1 TYPE=STIFFNESS
1 2.3E9 0. 0. 0. 0. 0. 0. -2.3E9 0. 0. 0. 0. 0.
2 0. 9.4E7 0. 0. 0. 0. 4.7E7 0. -9.4E7 0. 0. 0. 4.7E7
3 0. 0. 9.4E7 0. -4.7E7 0. 0. 0. -9.4E7 0. -4.7E7 0.
4 0. 0. 0. 7.3E6 0. 0. 0. 0. 0. -7.3E6 0. 0.
5 0. 0. 0. 0. 3.3E7 0. 0. 0. 4.7E7 0. 1.4E7 0.
6 0. 0. 0. 0. 0. 3.3E7 0. -4.7E7 0. 0. 0. 1.4E7
7 0. 0. 0. 0. 0. 0. 2.3E9 0. 0. 0. 0. 0.
8 0. 0. 0. 0. 0. 0. 0. 9.4E7 0. 0. 0. -4.7E7
9 0. 0. 0. 0. 0. 0. 0. 0. 9.4E7 0. 4.7E7 0.
10 0. 0. 0. 0. 0. 0. 0. 0. 0. 7.3E6 0. 0.
11 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 3.3E7 0.
12 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 3.3E7
*
EGROUP CONNECTOR NAME=1 SUBTYPE=MATRIX CONN-PROP=1
*
CONN-DEFINE NAME=1 GROUP=1 SIDE1TYPE=POINT SIDE1NAME=1,
    SIDE2TYPE=POINT TRIAD1OPT=AXIAL
CLEAR
2 0
DATAEND

```

Modeling example 5: Multilinear matrix element

The following example demonstrates the input for a multilinear matrix type connector element. The stiffness matrices are both symmetric (though they need not be), and there is no damping response. The multilinear elastic response depends upon the elastic force in the local 2-direction.

```

MATRIX-K NAME=1
1  2.5E5 -1.6E0 -2.7E+0 2.9E2  1.5E3 -1.1E3 -2.5E5  1.6E+0  2.7E+0 -2.9E2  6.1E2 -1.5E2
2  0.0    1.6E3 -3.5E-1 2.7E1  1.0E2  3.2E5  1.6E0 -1.6E+3  3.5E-1 -2.7E1  1.7E2  9.6E5
3  0.0    0.0    3.3E+2 1.5E1 -1.1E5 -9.8E1  2.7E0  3.5E-1 -3.3E+2 -1.5E1 -1.4E5 -1.8E2
4  0.0    0.0    0.0    5.5E6 -1.0E4  9.9E3 -2.9E2 -2.7E+1 -1.5E+1 -5.5E6 -1.5E3  1.1E4
5  0.0    0.0    0.0    0.0    5.2E7  3.9E4 -1.5E3 -1.0E+2  1.1E+5  1.0E4  3.6E7  4.3E4
6  0.0    0.0    0.0    0.0    0.0    1.1E8  1.1E3 -3.2E+5  9.8E+1 -9.9E3  3.6E4  1.3E8
7  0.0    0.0    0.0    0.0    0.0    0.0    2.5E5 -1.6E+0 -2.7E+0  2.9E2 -6.1E2  1.5E2
8  0.0    0.0    0.0    0.0    0.0    0.0    0.0    1.6E+3 -3.5E-1  2.7E1 -1.7E2 -9.6E5
9  0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    3.3E+2  1.5E1  1.4E5  1.8E2
10 0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    5.5E6  1.5E3 -1.1E4
11 0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    7.6E7  9.7E4
12 0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    6.1E8
*
MATRIX-K NAME=2
1  5.0E5 -2.2E0 -5.4E+0 5.8E2  3.0E3 -2.2E3 -5.0E5  3.2E+0  5.4E+0 -5.8E2  1.2E3 -3.0E2
2  0.0    2.2E3 -7.0E-1 5.4E1  2.0E2  6.4E5  3.2E0 -3.2E+3  7.0E-1 -5.4E1  3.4E2  2.0E6
3  0.0    0.0    6.6E+2 3.0E1 -2.2E5 -2.0E2  5.4E0  7.0E-1 -6.6E+2 -3.0E1 -2.8E5 -3.6E2
4  0.0    0.0    0.0    1.1E7 -2.0E4  2.0E4 -5.8E2 -5.4E+1 -3.0E+1 -1.1E7 -3.0E3  2.2E4
5  0.0    0.0    0.0    0.0    1.0E8  8.0E5 -3.0E3 -2.0E+2  2.2E+5  2.0E4  7.2E7  8.6E4
6  0.0    0.0    0.0    0.0    0.0    2.2E8  2.2E3 -6.4E+5  2.0E+2 -2.0E4  7.2E4  2.6E8
7  0.0    0.0    0.0    0.0    0.0    0.0    5.0E6 -3.2E+0 -5.4E+0  5.8E2 -6.1E2  3.0E2
8  0.0    0.0    0.0    0.0    0.0    0.0    0.0    3.2E+3 -7.0E-1  5.4E1 -3.4E2 -2.0E6
9  0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    6.6E+2  3.0E1  2.8E5  3.6E2
10 0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    1.1E7  3.0E3 -2.2E4
11 0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    1.5E8  2.0E5
12 0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    1.2E9
*
MATRIX-NL-K NAME=1 TYPE=ELASTIC-2 SYMMETRIC=YES LIMITS=EXTRAPOLATE
0.00      1
0.12      2
*
CONN-PROP MATRIX-MULTILINEAR KNL=1
*
EGROUP CONNECTOR SUBTYPE=MATRIX-MULTILINEAR CONN-PROP=1
*
CONN-DEFINE NAME=1 GROUP=1 SIDE1TYPE=POINT SIDE1NAME=1,
             SIDE2TYPE=POINT TRIAD1OPT=AXIAL
CLEAR
2 0
DATAEND

```

For simplicity, this example shows a multilinear response defined

using only two stiffness matrices, but each multilinear definition can use as many matrices as desired. Also, the multilinear matrix element may have both elastic and damping responses, and the responses can depend on different quantities. For example, the elastic response may depend on a damping force and the damping response can depend on the temperature.

2.14.4 Notes and Recommendations

- 1) The connector element cannot be used in explicit dynamic analysis or cyclic symmetry analysis.
- 2) In problems involving connector elements undergoing large rotations, it is recommended to use the nonsymmetric sparse solver (`MASTER SOLVER = NONSYM-SPARSE`).
- 3) In problems involving matrix type connector elements possessing nonsymmetric stiffness and/or damping matrices, the nonsymmetric sparse solver should be used.
- 4) In problems involving nonlinear joint type connector elements, it is recommended to use the line search feature (`ITERATION LINE-SEARCH = YES`).
- 5) If the user specifies damping behavior for connector elements in a static analysis, ADINA will emit the following warning message:


```
###Damping cannot be used in frequency or static solutions
```


ADINA will then proceed to perform the static analysis, ignoring the connector element's damping behavior. To model damping behavior in the connector element, the user must run an implicit dynamic analysis.
- 6) When a connector element is attached to a shell element, the attached node on the shell becomes a 6-DOF node which will resist an applied moment in the shell's drilling direction. For more information, see section 2.7.3 Shell nodal point degrees of freedom.
- 7) The default value for the backstop factor `BACKFAC` is `1.0E10`,

which may be too large in some instances. If a nonlinear joint element is experiencing convergence difficulties when reaching its backstop limit, consider using a smaller value for the backstop factor.

3. Material models and formulations

The objective of this chapter is to summarize the theoretical basis and practical use of the material models and formulations available in ADINA.

3.1 Stress and strain measures in ADINA

3.1.1 Summary

- It is important to recognize which stress and strain measures are employed in the use of a material model:
 - ▶ In the preparation of the input data in which the material parameters are defined with respect to these stress and strain measures
 - ▶ In the interpretation of the analysis results in which the type of stresses and strains output must be considered
- The practical use of the material models available in ADINA regarding the stress and strain measures used for the input data and analysis results is summarized in the following. These stress/strain measures are described in detail in ref. KJB, Section 6.2.
- In the following discussion, small strains are strains less than about 2%.

Small displacement/small strain formulation

Input of material parameters: All elements and material models use the engineering stress-engineering strain relationship.

Output: All elements and material models output engineering stresses and engineering strains.

Note that under small displacement, small strain conditions, engineering stresses and Cauchy stresses are nearly identical.

Large displacement/small strain formulation

Input of material parameters: 2nd Piola-Kirchhoff stresses and Green-Lagrange strains. Note that under small strain conditions, 2nd Piola-Kirchhoff stresses are nearly equal to engineering stresses, and Green-Lagrange strains are nearly equal to engineering strains.

Output:

- (1) 2-D, 3-D elements: all material models output Cauchy stresses and Green-Lagrange strains.
- (2) Beam, iso-beam, pipe elements: all material models output 2nd Piola-Kirchhoff stresses and Green-Lagrange strains.
- (3) Shell elements, results output in local coordinate system: all material models output 2nd Piola-Kirchhoff stresses and Green-Lagrange strains.
- (4) Shell elements, results output in global coordinate system: all material models output Cauchy stresses and rotated small strains.

Large displacement/large strain formulation, 2D and 3D elements

For the two- and three-dimensional solid elements, the following material models can be used:

- (1) Plastic-bilinear, plastic-multilinear, thermo-plastic, plastic-cyclic, thermo-plastic-cyclic. In these cases either the ULH formulation or the ULJ formulation is used. The default is the ULH formulation, except when explicit time integration is employed.

Input of material parameters: Cauchy (true) stresses and logarithmic (true) strains. For the multilinear stress-strain curves, it is also possible to enter engineering stress-strain data along with the input MASTER CONVERT-SSVAL=YES, see detailed description of the CONVERT-

SSVAL feature in Section 3.4.1.

Output: ULH formulation: Cauchy stresses and deformation gradients.

ULJ formulation: Cauchy stresses and Jaumann strains.

(2) Drucker-Prager, Mroz bilinear, creep, plastic-creep, multilinear-plastic-creep, plastic-creep-variable, multilinear-plastic-creep-variable, viscoelastic, user-supplied. In these cases the ULH formulation is always used.

Input of material parameters: Cauchy (true) stresses and logarithmic (true) strains

Output: Cauchy stresses and deformation gradients.

(3) Plastic-orthotropic. In this case the updated ULJ formulation is always used.

Input of material parameters: Cauchy (true) stresses and logarithmic (true) strains

Output: Cauchy stresses and Jaumann strains.

(4) Mooney-Rivlin, Ogden, Arruda-Boyce, hyper-foam, and Sussman-Bathe. In this case the TL (total Lagrangian) formulation is used.

Input of material parameters: Mooney-Rivlin, Ogden, Arruda-Boyce or hyper-foam constants. For Sussman-Bathe, the uniaxial stress-strain curve (see Section 3.8.1.5 for details).

Output: Cauchy stresses and deformation gradients.

Large displacement/large strain formulation, shell elements

For the shell elements, the following material models can be used, provided that the elements are single layer elements described using 3, 4, 6, 9 or 16 midsurface nodes:

(1) Plastic-bilinear, plastic-multilinear, plastic-cyclic, thermo-plastic-cyclic. Either the ULJ formulation or the ULH formulation can be used. The default is the ULH formulation, except when explicit time integration is employed.

(2) Plastic-orthotropic. The ULJ formulation is always employed.

Input of material parameters: Kirchhoff stresses and logarithmic (true) strains. For the multilinear stress-strain curves, it is also possible to enter engineering stress-strain data along with the input MASTER CONVERT-SSVAL=YES, see detailed description of the CONVERT-SSVAL feature in Section 3.4.1.

Output: Kirchhoff stresses and left Hencky strains (ULH) or Jaumann strains (ULJ).

Strain measures: The strain measures used in ADINA are illustrated hereafter in the simplified case of a rod under uniaxial tension (see Fig. 3.1-1).

Engineering strain:
$$e_0 = \frac{\ell - \ell_0}{\ell_0}$$

Green-Lagrange strain:
$$\varepsilon = \frac{1}{2} \frac{\ell^2 - \ell_0^2}{\ell_0^2}$$

Logarithmic strain, Hencky strain, Jaumann strain:

$$e = \ln \left(\frac{\ell}{\ell_0} \right) = \int_{\ell_0}^{\ell} \frac{d\ell}{\ell}$$

Stretch:
$$\lambda = \frac{\ell}{\ell_0}$$

ref. KJB
Sec. 6.2.2

- Green-Lagrange strains are used in the large displacement/small strain formulations. This is because large rotations do not affect the Green-Lagrange strains (Green-Lagrange strains are invariant with respect to rigid-body rotations), and for small strains, small rotations, Green-Lagrange strains and engineering strains are equivalent.

- In the small strain formulations, the current area is always assumed to be equal to the initial undeformed area.

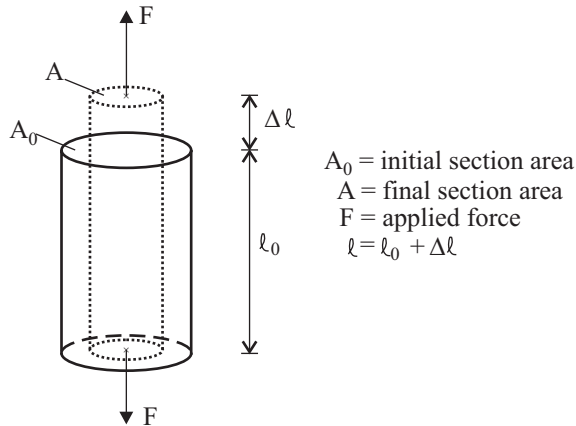


Figure 3.1-1: Rod under uniaxial tension

- Engineering strains are also called nominal strains in the literature.
- Logarithmic strains are also known as true strains.

Stress measures: The stress measures used in ADINA are illustrated hereafter in the simplified case of a rod under uniaxial tension (see Fig. 3.1-1).

Engineering stress:
$$\sigma = \frac{F}{A_0}$$

Cauchy stress:
$$\tau = \frac{F}{A} = \frac{\sigma A_0}{A}$$

2nd Piola-Kirchhoff stress:
$$S = \frac{F l_0}{A_0 l} = \frac{\sigma l_0}{l}$$

Kirchhoff stress:
$$J\tau = \frac{A\ell}{A_0\ell_0}\tau = \frac{F\ell}{A_0\ell_0} = \frac{\sigma\ell}{\ell_0}$$

- Cauchy stresses are also known as true stresses.

- For the case in which the material is incompressible,

$$\tau = J\tau = \frac{\sigma\ell}{\ell_0} \text{ can be used to compute the Cauchy stress and the}$$

Kirchhoff stress from the engineering stress.

- When the strains are small, the 2nd Piola-Kirchhoff stresses are nearly equal to the Cauchy stresses from which the rigid body rotations have been removed.

- 2nd Piola-Kirchhoff stresses are input only for the large displacement/small strain formulation. Because the strains are assumed to be small, the engineering stresses can be entered as the stress input quantities.

- 2nd Piola-Kirchhoff stresses are output only for the large displacement/small strain formulation and for element types in which the stresses expressed in the element coordinate system of the deformed element are of physical significance. If the element does not undergo rigid body rotations, the 2nd Piola-Kirchhoff stresses are nearly equal to the Cauchy stresses, because the strains are assumed to be small. If the element undergoes rigid body rotations, the Cauchy stresses expressed in the element coordinate system do not change (because the element coordinate system rotates). Since the 2nd Piola-Kirchhoff stresses, as expressed in the undeformed element coordinate system, do not change during a rigid body rotation, these 2nd Piola-Kirchhoff stresses remain nearly equal to the Cauchy stresses. The 2nd Piola-Kirchhoff stresses here provide a convenient way of removing the rigid body rotations from the output stresses.

- When the large displacement/small strain formulation is used, and when stresses and strains are output in an element local coordinate system, 2nd Piola-Kirchhoff stresses and Green-Lagrange strains are output. The justification for using 2nd Piola-Kirchhoff stresses is given in the preceding paragraph, and the

justification for using Green-Lagrange strains is similar.

- When the shell element is used with the large displacement/small strain formulation, and when stresses and strains are output in the global coordinate system, the stresses and strains are calculated in two steps. First the stresses and strains are calculated in the element local coordinate system, according to the preceding paragraph. Then the stresses and strains are rotated from the element local system into the global system. The resulting stresses can be interpreted as Cauchy stresses, because the strains are assumed small. The resulting strains can be interpreted as rotated small strains.
- When the material is nearly incompressible, the Kirchhoff stresses are nearly equal to the Cauchy stresses.
- Since Kirchhoff stresses are input/output only for materials that are nearly incompressible, practically speaking, the differences between Kirchhoff and Cauchy stresses are negligible.

3.1.2 Large strain thermo-plasticity analysis with the ULH formulation

- This section discusses the ULH formulation for large strain analysis. ULH stands for updated Lagrangian Hencky.
- The following is a quick summary of the theory of large strain inelastic analysis with the ULH formulation. For further information, see ref KJB, Section 6.6.4 and also the following references:

ref. F.J. Montáns and K.J. Bathe, "Computational issues in large strain elasto-plasticity: an algorithm for mixed hardening and plastic spin", *Int. J. Numer. Meth. Engng*, 2005; 63;159-196.

ref. M. Kojić and K.J. Bathe, *Inelastic Analysis of Solids and Structures*, Springer-Verlag, 2003.

Total deformation gradient tensor: Let \mathbf{X} be the total deformation gradient tensor at time t with respect to an initial

configuration taken at time 0. For ease of writing, we do not include the usual left superscripts and subscripts.

Polar decomposition into rotation and right stretch tensor: The total deformation gradient tensor \mathbf{X} can be decomposed into a material rigid-body rotation tensor \mathbf{R} and a symmetric positive-definite (right) stretch tensor \mathbf{U} (polar decomposition):

$$\mathbf{X} = \mathbf{R} \mathbf{U} \quad (3.1-1)$$

Principal directions of right stretch tensor: The right stretch tensor \mathbf{U} can be represented in its principal directions by a diagonal tensor $\mathbf{\Lambda}$, such that

$$\mathbf{U} = \mathbf{R}_L \mathbf{\Lambda} \mathbf{R}_L^T \quad (3.1-2)$$

where \mathbf{R}_L is a rotation tensor with respect to the fixed global axes (see Figure 3.1-2).

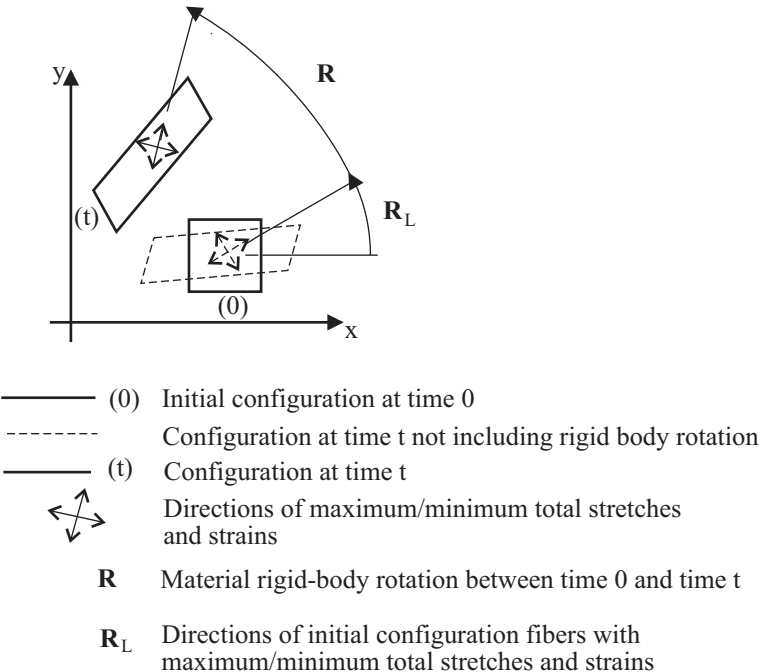


Figure 3.1-2: Directions of maximum/minimum total stretches and strains

(Note that the rotation \mathbf{R}_L does not correspond to a material rigid-body rotation, but to a rotation of the coordinate system: \mathbf{U} and $\mathbf{\Lambda}$ are two representations of the same deformed state, respectively in the global coordinate system and in the \mathbf{U} principal directions coordinate system.)

Right Hencky strain tensor: The Hencky strain tensor (computed in the right basis) is given by

$$\mathbf{E}^R = \ln \mathbf{U} = \mathbf{R}_I \ln \mathbf{\Lambda} \mathbf{R}_I^T \quad (3.1-3)$$

The superscript “R” symbolizes the right basis.

Polar decomposition into rotation and left stretch tensor: The total deformation gradient tensor \mathbf{X} can also be decomposed into a material rigid-body rotation \mathbf{R} and a symmetric positive-definite

(left) stretch tensor \mathbf{V} (polar decomposition):

$$\mathbf{X} = \mathbf{V} \mathbf{R} \quad (3.1-4)$$

\mathbf{R} in (3.1-4) is the same as \mathbf{R} in (3.1-1).

Principal directions of left stretch tensor: The left stretch tensor \mathbf{V} can be represented in its principal directions by a diagonal tensor $\mathbf{\Lambda}$, such that

$$\mathbf{V} = \mathbf{R}_E \mathbf{\Lambda} \mathbf{R}_E^T \quad (3.1-5)$$

where \mathbf{R}_E is a rotation tensor with respect to the fixed global axes.

Note that $\mathbf{R}_E = \mathbf{R} \mathbf{R}_L$.

Left Hencky strain tensor: The Hencky strain tensor (computed in the left basis) is given by

$$\mathbf{E}^L = \ln \mathbf{V} = \mathbf{R}_E \ln \mathbf{\Lambda} \mathbf{R}_E^T \quad (3.1-6)$$

The superscript “L” symbolizes the left basis.

Comparison of left and right Hencky strain tensors: The principal values of the left and right Hencky strain tensors are identical, and equal to the logarithms of the principal stretches. Hence both of these strain tensors can be considered to be logarithmic strain tensors. However, the principal directions of the left and right Hencky strain tensors are different. The principal directions of the right Hencky strain tensor do not contain the rigid body rotations of the material, but the principal directions of the left Hencky strain tensor contain the rigid body rotations of the material.

Therefore, for a material undergoing rigid body rotations, the principal directions of the right Hencky strain tensor do not rotate, however the principal directions of the left Hencky strain tensor rotate with the material. Hence, the left Hencky strain tensor is preferred for output and visualization of the strain state.

Multiplicative decomposition of deformation gradient in inelastic analysis: In inelastic analysis, the following

multiplicative decomposition of the total deformation gradient into an elastic deformation gradient \mathbf{X}^E and an inelastic deformation gradient \mathbf{X}^P is assumed:

$$\mathbf{X} = \mathbf{X}^E \mathbf{X}^P \quad (3.1-7)$$

To understand (3.1-7), consider a small region of material under a given stress state with deformation gradient \mathbf{X} . If this region of material is separated from the rest of the model and subjected to the same stress state, the deformation gradient is still \mathbf{X} . Now if the stress state is removed, (3.1-7) implies that the deformation gradient of the unloaded material is \mathbf{X}^P . The stresses are due entirely to the strains associated with the elastic deformation gradient \mathbf{X}^E .

It can be shown (see Montáns and Bathe), that (3.1-7) is equivalent to the additive decomposition of the displacements into elastic displacements and plastic displacements.

For the materials considered here, $\det \mathbf{X}^P = 1$.

Polar decomposition of elastic deformation gradient: The elastic deformation gradient can be decomposed into an elastic rotation tensor \mathbf{R}^E and elastic right and left stretch tensors \mathbf{U}^E , \mathbf{V}^E :

$$\mathbf{X}^E = \mathbf{R}^E \mathbf{U}^E = \mathbf{V}^E \mathbf{R}^E \quad (3.1-8-a,b)$$

Elastic Hencky strain tensors: The elastic Hencky strain tensors in the right and left bases are given by

$$\mathbf{E}^{ER} = \ln \mathbf{U}^E, \quad \mathbf{E}^{EL} = \ln \mathbf{V}^E \quad (3.1-9a,b)$$

Stress-strain relationships: The stresses are computed from the elastic Hencky strain tensors using the usual stress-strain law of isotropic elasticity. However, the stress measures used depend upon the strain measures used. When the right Hencky strain measure is used, the stress measure used is the rotated Kirchhoff stress

$$\bar{\boldsymbol{\tau}} = \left(\mathbf{R}^E \right)^T \mathbf{J} \boldsymbol{\tau} \mathbf{R}^E \quad (3.1-10)$$

and when the left Hencky strain measure is used, the stress measure is the (unrotated) Kirchhoff stress $\mathbf{J} \boldsymbol{\tau}$. $J = \det \mathbf{X}$ is the volume change of the material, and, using $\det \mathbf{X}^P = 1$, $J = \det \mathbf{X}^E$.

With these choices of stress and strain measures, the stresses and strains are work-conjugate.

The choice of right Hencky strain and rotated Kirchhoff stresses gives the same numerical results as the choice of left Hencky strain and (unrotated) Kirchhoff stresses.

Implementation notes: For 2-D and 3-D solid elements, the difference between the Cauchy and Kirchhoff stresses is neglected. The stress measure used with the right Hencky strains is

$\bar{\boldsymbol{\tau}} = \left(\mathbf{R}^E \right)^T \boldsymbol{\tau} \mathbf{R}^E$. The input of material properties is assumed to be in terms of Cauchy stresses, and the output of stresses is in terms of Cauchy stresses.

For shell elements, Kirchhoff stresses are used throughout. The input of material properties is assumed to be in terms of Kirchhoff stresses, and the output of stresses is in terms of Kirchhoff stresses.

These assumptions are justified because they are used with material models in which the plastic deformations are incompressible and the plastic deformations are generally much larger than the elastic deformations.

3.1.3 Large strain thermo-plasticity analysis with the ULJ formulation

- This section discusses the ULJ formulation for large strain inelastic analysis (ULJ formulation). ULJ stands for updated Lagrangian Jaumann.
- The following is a quick summary of the theory of large strain inelastic analysis with the ULJ formulation:

For further information, see ref KJB, Section 6.2.2 and also the

following reference:

ref. M. Kojić and K.J. Bathe, *Inelastic Analysis of Solids and Structures*, Springer-Verlag, 2003.

Velocity gradient tensor: The velocity gradient tensor is defined as

$$\mathbf{L} = \left[\frac{\partial' \dot{u}_i}{\partial' x_j} \right] = \dot{\mathbf{X}} \mathbf{X}^{-1} \quad (3.1-11)$$

Notice that the derivative is taken with respect to the current coordinates.

Rate of deformation tensor, spin tensor: The rate of deformation tensor is defined as

$$\mathbf{D} = \frac{1}{2} (\mathbf{L} + \mathbf{L}^T) \quad (3.1-12)$$

and the spin tensor is defined as

$$\mathbf{W} = \frac{1}{2} (\mathbf{L} - \mathbf{L}^T) \quad (3.1-13)$$

\mathbf{D} is the symmetric part of \mathbf{L} and \mathbf{W} is the skew-symmetric part of \mathbf{L} .

Rate of change of Jaumann strain tensor: The rate of change of the Jaumann strain is defined as

$$\dot{\boldsymbol{\varepsilon}}^J = \mathbf{D} + \mathbf{W} \boldsymbol{\varepsilon}^J - \boldsymbol{\varepsilon}^J \mathbf{W} \quad (3.1-14)$$

The quantity $\boldsymbol{\varepsilon}^J$ is termed the Jaumann strain in analogy with the more often-used Jaumann stress. But we do not use the Jaumann stress in the ULJ formulation.

Jaumann strain tensor: In practice, increments are used in computing the Jaumann strain tensor, i.e.

$${}^{t+\Delta t}\boldsymbol{\varepsilon}^J = {}^t\boldsymbol{\varepsilon}^J + (\mathbf{D}\Delta t) + (\mathbf{W}\Delta t) {}^t\boldsymbol{\varepsilon}^J - {}^t\boldsymbol{\varepsilon}^J (\mathbf{W}\Delta t) \quad (3.1-15)$$

Comparison of Jaumann strain with left Hencky strain: When the rate of change of the principal directions of the left stretch tensor \mathbf{V} is zero, the rate of change of the left Hencky strain is the same as the rate of change of the Jaumann strain. Hence the Jaumann strain can be used as an approximate replacement for the left Hencky strain. The Jaumann strain can be computed more efficiently than the left Hencky strain, because it is not necessary to take the square root or logarithm of a tensor when computing the Jaumann strain. On the other hand, the time step size affects the Jaumann strain, so that finite time step sizes lead to an error in the calculation of the Jaumann strain.

For a uniaxial deformation, the Jaumann strain approaches the logarithmic strain as the step size is reduced. For a rigid-body rotation, the Jaumann strain also rotates, with the rotation of the Jaumann strain approaching the expected rotation as the step size is reduced.

It can also be shown that the Jaumann strain is path-dependent in general, so that a deformation history in which the final deformations equal the initial deformations can produce (non-physical) non-zero Jaumann strains, even in the limit of infinitesimally small time steps.

Stress-strain relationships: In elasto-plasticity, the same algorithms are used as in small-strain elasto-plasticity. The mechanical strains are computed as the total strains minus the plastic strains (and also any thermal strains), in which the total strains are the Jaumann strains.

As in the ULH formulation, the stresses are Cauchy stresses for 2-D / 3-D elements, and are Kirchhoff stresses for shell elements.

3.1.4 Thermal strains

- Calculation of thermal strains is needed for temperature-dependent material models, as well as temperature-invariant material models with non-zero thermal expansion coefficients.

- The temperature at an integration point is evaluated based on the nodal temperatures and the element shape functions, and then used to calculate the thermal strains.
- For isotropic temperature independent materials, the following expression is used for thermal expansion.

$${}^t e_{ij}^{TH} = \alpha \left({}^t \theta - {}^0 \theta \right) \delta_{ij} \quad (3.1-16)$$

where δ_{ij} is the Kronecker delta ($\delta_{ij} = 1$ for $i = j$ and $\delta_{ij} = 0$ for $i \neq j$).

- If the thermal expansion is temperature dependent and isotropic, the thermal strains are calculated as follows:

$${}^t e_{ij}^{TH} = {}^t \bar{\alpha} \left({}^t \theta - {}^0 \theta \right) \delta_{ij} \quad (3.1-17)$$

where

$${}^t \bar{\alpha} = \frac{1}{\left({}^t \theta - {}^0 \theta \right)} \left(\alpha \left({}^t \theta \right) \left({}^t \theta - \theta_{REF} \right) - \alpha \left({}^0 \theta \right) \left({}^0 \theta - \theta_{REF} \right) \right) \quad (3.1-18)$$

and θ_{REF} is the material reference temperature and is input as a material property, see below for more details.

- For temperature independent orthotropic materials Eq. (3.1-16) is replaced by a thermal expansion coefficient vector,

$${}^t e_{ij}^{TH} = \alpha_i \left({}^t \theta - {}^0 \theta \right) \delta_{ij} \quad (\text{no summation over } i) \quad (3.1-19)$$

- For temperature dependent orthotropic materials Eq. (3.1-17) and Eq. (3.1-18) are modified for each direction similar to Eq. (3.1-19).
- Equations (3.1-17) and (3.1-18) are derived as follows: Suppose that, from experimental data, the dependence of the length of a bar as a function of temperature is obtained, as shown in Fig. 3.1-3.

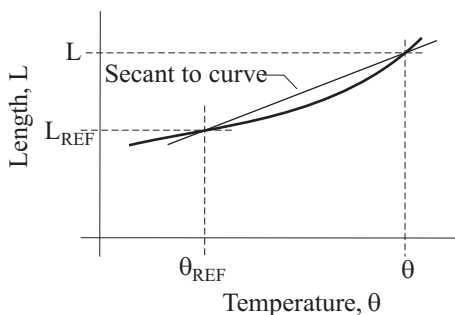


Figure 3.1-3: Length of bar vs. temperature

The thermal strain with respect to the reference length may be calculated as

$$\varepsilon^{TH} = \frac{L - L_{REF}}{L_{REF}}$$

Then we define the mean coefficient of thermal expansion for a given temperature as follows:

$$\alpha(\theta) = \frac{\varepsilon^{TH}(\theta)}{\theta - \theta_{REF}}$$

With this definition, the secant slope in Fig. 3.1-3 is $L_{REF} \alpha(\theta)$. Now, in ADINA, we assume that the thermal strains are initially zero. To do this, we subtract the thermal strain corresponding to ${}^0\theta$ to obtain

$${}^t\varepsilon^{TH} = \alpha({}^t\theta)({}^t\theta - \theta_{REF}) - \alpha({}^0\theta)({}^0\theta - \theta_{REF})$$

which leads to Equations (3.1-17) and (3.1-18).

Notice that if the mean coefficient of thermal expansion is constant, θ_{REF} no longer enters into the definition of ${}^t\bar{\alpha}$ and the equations reduce to Eq. 3.1-16. In general, when the mean coefficient of thermal expansion is not constant, θ_{REF} must be chosen based on

knowledge of the experiment used to determine $\alpha(\theta)$ since for the same material curve, different choices of θ_{REF} yield different values of $\alpha(\theta)$.

3.2 Linear elastic material models

- The following material models are discussed in this section:

Elastic-isotropic: isotropic linear elastic

Elastic-orthotropic: orthotropic linear elastic

In each model, the total stress is uniquely determined by the total strain.

- These models can be employed using the **small displacement** or **large displacement** formulations. In all cases, the strains are assumed small.

When the elastic-isotropic and elastic-orthotropic materials are used with the small displacement formulation, the formulation is linear.

- If the material models are employed with a large displacement/small strain or large displacement/large strain formulations, the total Lagrangian formulation is used.
- In the small displacement formulation, the stress-strain relationship is

$${}^t_0\sigma = \mathbf{C} {}^t_0\mathbf{e}$$

in which ${}^t_0\sigma$ = engineering stresses and ${}^t_0\mathbf{e}$ = engineering strains.

- In the total Lagrangian formulation, the stress-strain relationship is

$${}^t_0\mathbf{S} = \mathbf{C} {}^t_0\boldsymbol{\varepsilon}$$

in which ${}^t_0\mathbf{S}$ = second Piola-Kirchhoff stresses and ${}^t_0\boldsymbol{\varepsilon}$ = Green-Lagrange strains.

ref. KJB
Section 6.6.1

- The same matrix **C** is employed in all of these formulations.
- If the strains are large, it is recommended that the linear elastic material model not be used (see ref KJB, pp 589-590 for a discussion in which a linear elastic material is used under large strain conditions).

3.2.1 Elastic-isotropic material model

- This material model is available for the **truss, 2-D solid, 3-D solid, beam, iso-beam, plate, shell** and **pipe** elements.
- The two material constants used to define the constitutive relation (the matrix **C**) are

E = Young's modulus, ν = Poisson's ratio

ref. KJB
Table 4.3,
p. 194

- The same constants are employed in the small and large displacement formulations, and hence the matrices **C** are identically the same in all formulations.
- You can specify the coefficient of thermal expansion as part of the elastic-isotropic material description. The coefficient of thermal expansion is assumed to be temperature-independent.

3.2.2 Elastic-orthotropic material model

- The elastic-orthotropic material model is available for the **2-D solid, 3-D solid, plate** and **shell** elements.

2-D solid elements: Figure 3.2-1 shows a typical two-dimensional element, for which the in-plane orthogonal material axes are "a" and "b". The third orthogonal material direction is "c" and is perpendicular to the plane defined by "a" and "b". The material constants are defined in the principal material directions (a,b,c).

- You can specify the coefficients of thermal expansion as part of the elastic-isotropic material description. The coefficients of

thermal expansion are assumed to be temperature-independent.

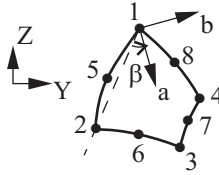


Figure 3.2-1: Principal in-plane material axes orientation for the orthotropic material model for 2-D solid elements

The stress-strain matrices are as follows for the various 2-D element types:

Plane stress:

$$\begin{bmatrix} e_a \\ e_b \\ e_c \\ \gamma_{ab} \end{bmatrix} = \begin{bmatrix} \frac{1}{E_a} & -\frac{\nu_{ab}}{E_b} & -\frac{\nu_{ac}}{E_c} & 0 \\ -\frac{\nu_{ba}}{E_a} & \frac{1}{E_b} & -\frac{\nu_{bc}}{E_c} & 0 \\ -\frac{\nu_{ca}}{E_a} & -\frac{\nu_{cb}}{E_b} & \frac{1}{E_c} & 0 \\ 0 & 0 & 0 & \frac{1}{G_{ab}} \end{bmatrix} \begin{bmatrix} \sigma_a \\ \sigma_b \\ \sigma_c = 0 \\ \sigma_{ab} \end{bmatrix}$$

Plane strain:

$$\begin{bmatrix} e_a \\ e_b \\ e_c = 0 \\ \gamma_{ab} \end{bmatrix} = \begin{bmatrix} \frac{1}{E_a} & -\frac{\nu_{ab}}{E_b} & -\frac{\nu_{ac}}{E_c} & 0 \\ -\frac{\nu_{ba}}{E_a} & \frac{1}{E_b} & -\frac{\nu_{bc}}{E_c} & 0 \\ -\frac{\nu_{ca}}{E_a} & -\frac{\nu_{cb}}{E_b} & \frac{1}{E_c} & 0 \\ 0 & 0 & 0 & \frac{1}{G_{ab}} \end{bmatrix} \begin{bmatrix} \sigma_a \\ \sigma_b \\ \sigma_c \\ \sigma_{ab} \end{bmatrix}$$

Axisymmetric:

$$\begin{bmatrix} e_a \\ e_b \\ e_c \\ \gamma_{ab} \end{bmatrix} = \begin{bmatrix} \frac{1}{E_a} & -\frac{\nu_{ab}}{E_b} & -\frac{\nu_{ac}}{E_c} & 0 \\ -\frac{\nu_{ba}}{E_a} & \frac{1}{E_b} & -\frac{\nu_{bc}}{E_c} & 0 \\ -\frac{\nu_{ca}}{E_a} & -\frac{\nu_{cb}}{E_b} & \frac{1}{E_c} & 0 \\ 0 & 0 & 0 & \frac{1}{G_{ab}} \end{bmatrix} \begin{bmatrix} \sigma_a \\ \sigma_b \\ \sigma_c \\ \sigma_{ab} \end{bmatrix}$$

The seven material constants (E_a , E_b , E_c , ν_{ab} , ν_{ac} , ν_{bc} and G_{ab}) define the symmetric compliance matrix \mathbf{C}_ℓ^{-1} .

$$\mathbf{C}_\ell^{-1} = \begin{bmatrix} \frac{1}{E_a} & -\frac{\nu_{ab}}{E_b} & 0 & -\frac{\nu_{ac}}{E_c} \\ & \frac{1}{E_b} & 0 & -\frac{\nu_{bc}}{E_c} \\ & & \frac{1}{G_{ab}} & 0 \\ \text{symmetric} & & & \frac{1}{E_c} \end{bmatrix}$$

where the subscript " ℓ " in \mathbf{C}_ℓ^{-1} indicates that the material law is given in the local system of the material axes. Note that the determinant of \mathbf{C}_ℓ^{-1} must not be zero in order to be able to calculate the inverse \mathbf{C}_ℓ . This imposes the following restrictions on the constants. The material constants must be defined so that the stress-strain constitutive matrix is positive-definite; i.e.,:

$$|\nu_{ji}| < \left(\frac{E_i}{E_j} \right)^{\frac{1}{2}}, \quad i, j = a, b, c$$

$$\nu_{ab}\nu_{bc}\nu_{ca} < 0.5 \left(1 - \nu_{ab}^2 \frac{E_a}{E_b} - \nu_{bc}^2 \frac{E_b}{E_c} - \nu_{ca}^2 \frac{E_c}{E_a} \right) \leq 0.5$$

or

$$\nu_{ab}\nu_{bc}\nu_{ac} \frac{E_a}{E_c} < 0.5 \left(1 - \nu_{ab}^2 \frac{E_a}{E_b} - \nu_{bc}^2 \frac{E_b}{E_c} - \nu_{ac}^2 \frac{E_a}{E_c} \right) \leq 0.5$$

using ADINA input values.

Based on the input values for ν_{ij} , ADINA calculates ν_{ji} so as to have a symmetric constitutive matrix; i.e.,

$$\frac{\nu_{ji}}{E_i} = \frac{\nu_{ij}}{E_j}$$

See also for example the following reference:

ref. Jones, R.M., *Mechanics of Composite Materials*, McGraw-Hill p. 42, 1975.

In ADINA, the Poisson's ratio ν_{ij} is defined differently from Jones, although it is consistent with Jones' nomenclature. In Figure 3.2-2, a uniaxial tensile test is illustrated in which the material under test is loaded in the a -direction. The Poisson's ratio is defined in ADINA by

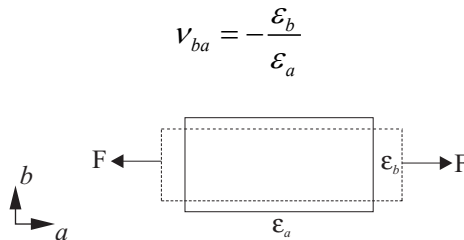


Figure 3.2-2: Definition of ν_{ba}

In general,

$$\nu_{ij} = -\frac{\varepsilon_i}{\varepsilon_j} \quad (i, j = a, b, c)$$

In Jones, however, the same Poisson's ratio is notated

$$^J\nu_{ij} = -\frac{\varepsilon_j}{\varepsilon_i}.$$

Using the correspondence between the Jones and ADINA nomenclature for the Poisson's ratio,

$$\nu_{ji} = ^J\nu_{ij}$$

and the relationship

$$\frac{\nu_{ji}}{E_i} = \frac{\nu_{ij}}{E_j}$$

it is possible to calculate the value of ν_{ji} given a value for $^J\nu_{ij}$ from

$$\nu_{ij} = \left(\frac{E_j}{E_i} \right) ^J\nu_{ij}$$

When material data is available from sources that follow Jones' notation of Poisson's ratio, the above conversion allows the equivalent ADINA Poisson's ratio to be calculated from the Jones Poisson's ratio for input to ADINA.

To obtain the stress-strain matrix \mathbf{C} corresponding to the global material axes, first \mathbf{C}_ℓ is calculated and then

$$\mathbf{C} = \mathbf{Q}^T \mathbf{C}_\ell \mathbf{Q}$$

where

$$\mathbf{Q} = \begin{bmatrix} \ell_1^2 & m_1^2 & \ell_1 \ell_2 & 0 \\ \ell_2^2 & m_2^2 & m_1 m_2 & 0 \\ 2m_1 m_2 & 2\ell_1 \ell_2 & \ell_1 m_2 + \ell_2 m_1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and ℓ_1, ℓ_2 and m_1, m_2 are the direction cosines of the a, b material axes to the global axes.

Note that in a large displacement analysis it is assumed that the angle β remains constant throughout the incremental solution. Hence, the use of the total Lagrangian formulation is usually most appropriate.

3-D solid elements: In three-dimensional analysis we have

$$\mathbf{C}_\ell^{-1} = \begin{bmatrix} 1/E_a & -\nu_{ab}/E_b & -\nu_{ac}/E_c & 0 & 0 & 0 \\ & 1/E_b & -\nu_{bc}/E_c & 0 & 0 & 0 \\ & & 1/E_c & 0 & 0 & 0 \\ & & & 1/G_{ab} & 0 & 0 \\ & & & & 1/G_{ac} & 0 \\ \text{symmetric} & & & & & 1/G_{bc} \end{bmatrix}$$

and, as in two-dimensional analysis, this matrix is inverted and then transformed to the global coordinate axes.

Plate elements: Figure 3.2-3 shows a typical plate element for which the in-plane orthogonal material axes are "a" and "b". The material constants are defined in the principal material directions (a,b), for which we have

$$\begin{bmatrix} e_a \\ e_b \\ e_c \\ \gamma_{ab} \end{bmatrix} = \begin{bmatrix} 1/E_a & -\nu_{ab}/E_b & -\nu_{ac}/E_c & 0 \\ -\nu_{ba}/E_a & 1/E_b & -\nu_{bc}/E_c & 0 \\ -\nu_{ca}/E_a & -\nu_{cb}/E_b & 1/E_c & 0 \\ 0 & 0 & 0 & 1/G_{ab} \end{bmatrix} \begin{bmatrix} \sigma_a \\ \sigma_b \\ \sigma_c = 0 \\ \sigma_{ab} \end{bmatrix}$$

Note that the transverse normal strain e_c is not calculated for the plate elements.

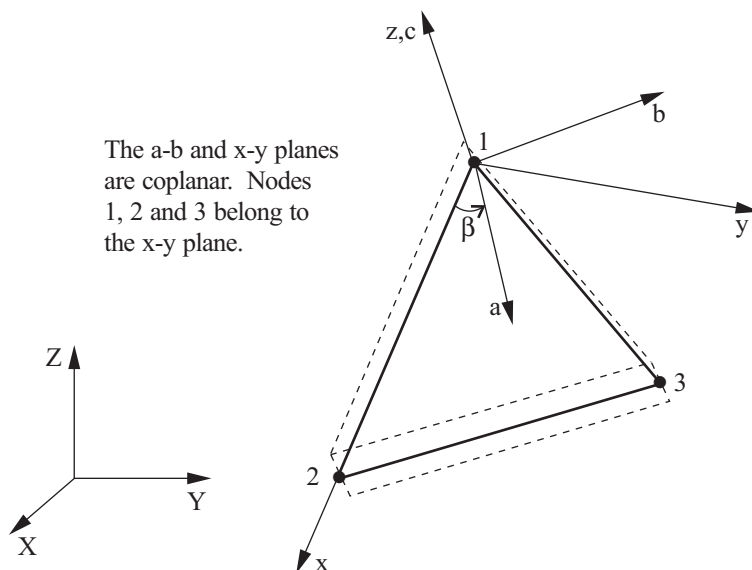


Figure 3.2-3: Principal in-plane material axes orientation for the orthotropic material model for plate elements

In order to obtain the stress-strain matrix \mathbf{C} corresponding to the local x y z directions of the element, we use

$$\mathbf{C} = \mathbf{Q}^T \mathbf{C}_\ell \mathbf{Q}^T$$

$$\text{where } \mathbf{Q}^T = \begin{bmatrix} \cos\beta \cos\beta & \sin\beta \sin\beta & \cos\beta \sin\beta \\ \sin\beta \sin\beta & \cos\beta \cos\beta & -\cos\beta \sin\beta \\ -2\cos\beta \sin\beta & 2\cos\beta \sin\beta & \cos^2\beta - \sin^2\beta \end{bmatrix}$$

β is the angle between the a and x axes, see Fig. 3.2-2.

Shell elements: Figure 3.2-4 shows typical shell elements for which the orthogonal material axes are " a ", " b " and " c ".

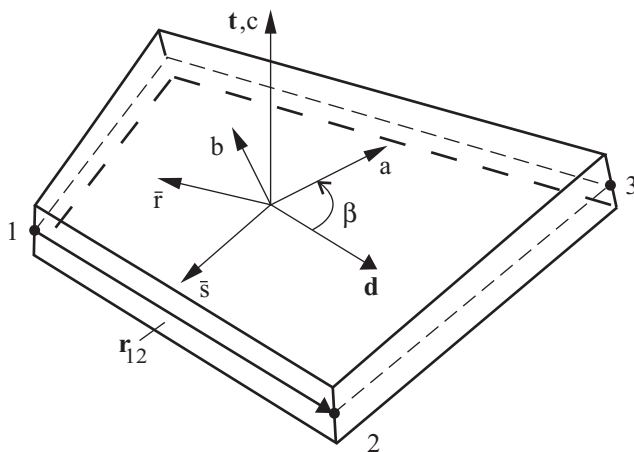
The constitutive relation defined in the a,b,c system is

$$\begin{bmatrix} e_a \\ e_b \\ e_c \\ \gamma_{ab} \\ \gamma_{ac} \\ \gamma_{bc} \end{bmatrix} = \begin{bmatrix} 1/E_a & -\nu_{ab}/E_b & -\nu_{ac}/E_c & 0 & 0 & 0 \\ -\nu_{ba}/E_a & 1/E_b & -\nu_{bc}/E_c & 0 & 0 & 0 \\ -\nu_{ca}/E_a & -\nu_{cb}/E_b & 1/E_c & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/G_{ab} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/G_{ac} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/G_{bc} \end{bmatrix} \begin{bmatrix} \sigma_a \\ \sigma_b \\ \sigma_c = 0 \\ \sigma_{ab} \\ \sigma_{ac} \\ \sigma_{bc} \end{bmatrix}$$

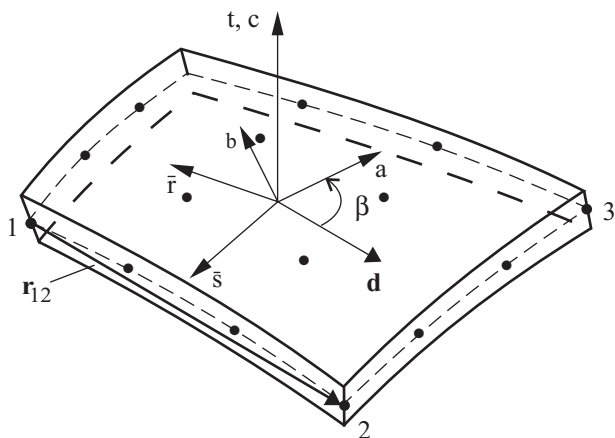
- Table 3.2-1 summarizes which material constants are necessary, depending on the type of element used in the analysis.

Table 3.2-1: Required material constants for the element types available in ADINA

Element type	Required material constants (these constants must be nonzero except for Poisson's ratios)
2-D solid	$E_a, E_b, E_c, \nu_{ab}, \nu_{ac}, \nu_{bc}, G_{ab}$
3-D solid	$E_a, E_b, E_c, \nu_{ab}, \nu_{ac}, \nu_{bc}, G_{ab}, G_{ac}, G_{bc}$
Plate	$E_a, E_b, \nu_{ab}, G_{ab}$
Shell	$E_a, E_b, E_c, \nu_{ab}, \nu_{ac}, \nu_{bc}, G_{ab}, G_{ac}, G_{bc}$



(a) 4-node element



(b) 16-node element

Note: (\bar{r}, \bar{s}, t) is the local Cartesian system.
 The a - b and \bar{r} - \bar{s} planes are coplanar.
 \mathbf{d} is the unit projection of \mathbf{r}_{12} onto the \bar{r} - \bar{s} plane.
 β is the input material angle.

Figure 3.2-4: Definition of axes of orthotropy for shell elements

3.3 Nonlinear elastic material model

- The nonlinear elastic material model is available for truss, 2-D solid (plane stress, plane strain and axisymmetric), 3-D solid and shell elements. The formulations used are slightly different (and simpler) for the truss element, as detailed in Section 3.3.1.
- This material uses a nonlinear elastic uniaxial stress-strain data input in tabular form and shown in Fig. 3.3-1. Note that the input curve must pass through the origin (0,0). This material is not based on the classical theory of finite elasticity, and is not intended for large strain analysis. However, it is a useful material model when used appropriately, and with awareness of its limitations. This material model is also available as a user-coded material for 2-D and 3-D solids only.
- Note that the material unloads along the same curve, so that no permanent inelastic strains are introduced.
- The material can have different stress-strain curves in tension and compression. Under predominantly uniaxial tension or compression, the material response will follow the input curve exactly. Under shear dominated loading, the stress is interpolated from both tension and compression parts of the material stress-strain curve.

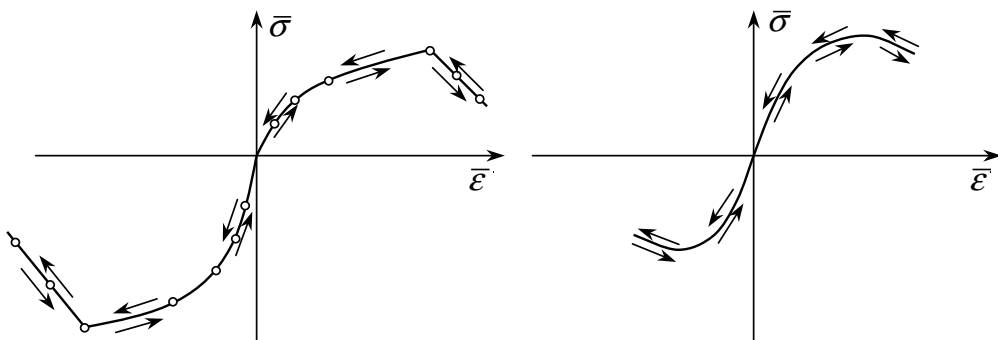


Figure 3.3-1: Stress-strain behavior of nonlinear elastic material model

- In order to use the uniaxial stress-strain data $\bar{\sigma}(\bar{\epsilon})$ of Fig. 3.3-1, the effective stress and strain ($\bar{\sigma}$ and $\bar{\epsilon}$) must be calculated based on the 2-D or 3-D total stress and strain tensors ($\boldsymbol{\sigma}$ and $\boldsymbol{\epsilon}$). The von-Mises stress is used as the effective stress, while the effective strain is based on

$$\int \bar{\sigma} d\bar{\epsilon} = \int \boldsymbol{\sigma}^T d\boldsymbol{\epsilon} \quad (3.3-1)$$

which equates the deformation work per unit volume in uniaxial loading to the multi-dimensional state. This results in a unique equation for $\bar{\epsilon}$ as a function of $\boldsymbol{\epsilon}$, ν and the stress-strain state that depends on the element type.

- The effective strain, $\bar{\epsilon}$, is defined by

$$\frac{1}{2} E_0 \bar{\epsilon}^2 = \frac{1}{2} \boldsymbol{\epsilon}^T \mathbf{C}_0 \boldsymbol{\epsilon} \quad (3.3-2)$$

where E_0 is Young's modulus which is determined by the most stiff region of the input stress-strain curve, \mathbf{C}_0 is the elastic stress-strain matrix obtained using E_0 and ν . (E_0 cancels out from both sides of Eq. (3.3-2))

Differentiating Eq. (3.3-2) with respect to the total strain, we have

$$d\bar{\epsilon} = \frac{1}{E_0 \bar{\epsilon}} \boldsymbol{\epsilon}^T \mathbf{C}_0 d\boldsymbol{\epsilon} \quad (3.3-3)$$

Substituting Eq. (3.3-3) into Eq. (3.3-1), the stresses can be expressed in terms of total strains, i.e.,

$$\boldsymbol{\sigma} = \frac{\bar{\sigma}}{E_0 \bar{\epsilon}} \mathbf{C}_0 \boldsymbol{\epsilon} \quad (3.3-4)$$

or

$$\boldsymbol{\sigma} = \frac{\bar{\sigma}}{E_0 \bar{\varepsilon}} \boldsymbol{\sigma}_e \quad (3.3-5)$$

where $\boldsymbol{\sigma}_e = \mathbf{C}_0 \boldsymbol{\varepsilon}$ which is the elastic trial stress.

- The effective stress $\bar{\sigma}$ is taken from the tensile part of the stress-strain curve for predominantly tensile loading, from the compression part of the stress-strain curve for predominantly compression loading and is interpolated between the two curves otherwise.
- The consistent tangent stress-strain matrix is obtained by differentiating Eq. (3.3-4) or (3.3-5) with respect to the total strain tensor. The stress-strain matrix is symmetric in predominant tensile or compression loading (when only one of the two material curves is used), and is non-symmetric otherwise (when interpolation between the curves is required). If a symmetric solver is used, the constitutive matrix is symmetrized and in most cases reasonable convergence is still obtained.
- Note that discontinuities are not allowed in user-supplied stress-strain curve. The table look-up is performed using linear interpolation within the table and linear extrapolation outside the table using the two starting or ending points.

Stress update algorithm

For an iteration i , given ${}^t \boldsymbol{\sigma}$, ${}^t \boldsymbol{\varepsilon}$, ${}^{t+\Delta t} \mathbf{u}^{(i)}$, E_0 , ν , update ${}^{t+\Delta t} \boldsymbol{\sigma}^{(i)}$, ${}^{t+\Delta t} \boldsymbol{\varepsilon}^{(i)}$

Step 1. Calculate the new total strain state ${}^{t+\Delta t} \boldsymbol{\varepsilon}^{(i)}$ based on displacements ${}^{t+\Delta t} \mathbf{u}^{(i)}$

Step 2. Calculate the elastic trial stress,

$${}^{t+\Delta t} \boldsymbol{\sigma}_e = \mathbf{C}_0 {}^{t+\Delta t} \boldsymbol{\varepsilon} \quad (3.3-6)$$

Step 3. Compute the magnitude of the effective strain, $\bar{\varepsilon}$.

Step 4. Calculate the ratio

$$r = C \frac{I_{1_e}}{^{t+\Delta t} \bar{\sigma}_e} \quad (3.3-7)$$

where C is a constant that biases the general stress state towards the pure tension or compression curves and is internally set to 3/2, I_{1_e} is the first elastic stress invariant, and $^{t+\Delta t} \bar{\sigma}_e$ is the effective elastic stress which is calculated as follows,

$$^{t+\Delta t} \bar{\sigma}_e = E_0 \ ^{t+\Delta t} \bar{\varepsilon} \quad (3.3-8)$$

Restrict r to be between -1 and 1.

Step 5. Calculate the effective stress in tension $\bar{\sigma}_t$ and in compression $\bar{\sigma}_c$, based on the user-supplied stress-strain curve and $\bar{\varepsilon}$, as follows:

$$^{t+\Delta t} \bar{\sigma}_t = ^{t+\Delta t} \bar{\sigma} (^{t+\Delta t} \bar{\varepsilon}) \quad (3.3-9)$$

$$^{t+\Delta t} \bar{\sigma}_c = - ^{t+\Delta t} \bar{\sigma} (- ^{t+\Delta t} \bar{\varepsilon}) \quad (3.3-10)$$

Step 6. Calculate the actual effective stress, $^{t+\Delta t} \bar{\sigma}$, as

$$^{t+\Delta t} \bar{\sigma} = \frac{1+r}{2} \ ^{t+\Delta t} \bar{\sigma}_t + \frac{1-r}{2} \ ^{t+\Delta t} \bar{\sigma}_c \quad (3.3-11)$$

Step 7. Evaluate the new stress state by

$$^{t+\Delta t} \boldsymbol{\sigma} = \frac{^{t+\Delta t} \bar{\sigma}}{E_0 \ ^{t+\Delta t} \bar{\varepsilon}} \ ^{t+\Delta t} \boldsymbol{\sigma}_e \quad (3.3-12)$$

Step 8. Evaluate the tangential stress-strain matrix. Symmetrize it if necessary.

Step 9. A secant stress-strain matrix can be used if the tangential stress-strain matrix exhibits a softening behavior.

3.3.1 Nonlinear elastic material for truss element

- For the truss element, an elastic material model is available for which the stress-strain relationship is defined as piecewise linear. Fig. 3.3-2 illustrates the definition of the stress-strain law.

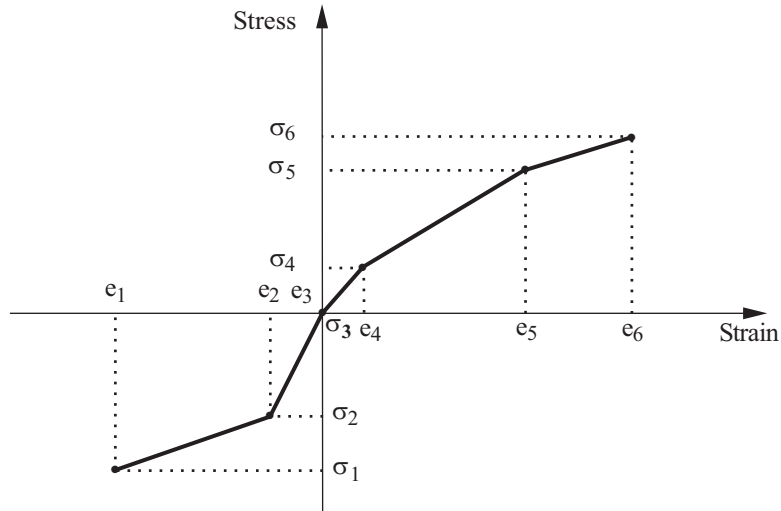


Figure 3.3-2: Nonlinear elastic material for truss element

Note that the stress is uniquely defined as a function of the strain only; hence for a specific strain ϵ , reached in loading or unloading, a unique stress is obtained from the curve in Fig. 3.3-2.

- A typical example of the truss nonlinear elastic model is shown in Fig. 3.3-3. Note that the stress-strain relation shown corresponds to a cable-like behavior in which the truss supports tensile but no compressive loading.

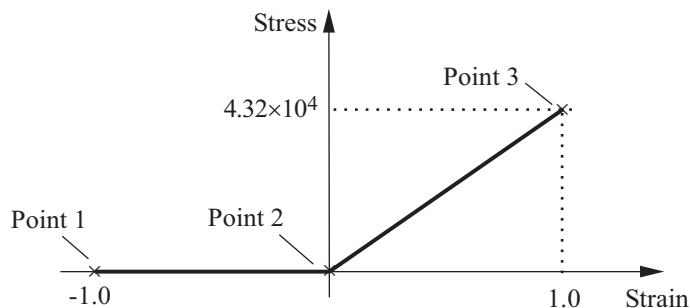


Figure 3.3-3: Example of nonlinear elastic stress-strain material model for truss

A sufficient range (in terms of the strain) must be used in the definition of the stress-strain relation so that the element strain evaluated in the solution lies within that range; i.e., referring to Fig. 3.3-2, we must have $e_1 \leq e \leq e_6$ for all t .

- The truss element with this material model is particularly useful in modeling gaps between structures. This modeling feature is illustrated in Fig. 3.3-4. Note that to use the gap element, it is necessary to know which node of one body will come into contact with which node of the other body.

ref. S.M. Ma and K.J. Bathe, "On Finite Element Analysis of Pipe Whip Problems," *Nuclear Engineering and Design*, Vol. 37, pp. 413-430, 1976.

ref. K.J. Bathe and S. Gracewski, "On Nonlinear Dynamic Analysis using Substructuring and Mode Superposition," *Computers and Structures*, Vol. 13, pp. 699-707, 1981.

Note that an alternate gap element can be obtained by the use of the 2-node truss element employing the plastic-bilinear material model, see Section 3.4.1.

A more general way of modeling contact between bodies is the use of contact surfaces, see Chapter 4.

- Other modeling features available with the truss element and this material model are shown in Figure 3.3-5.

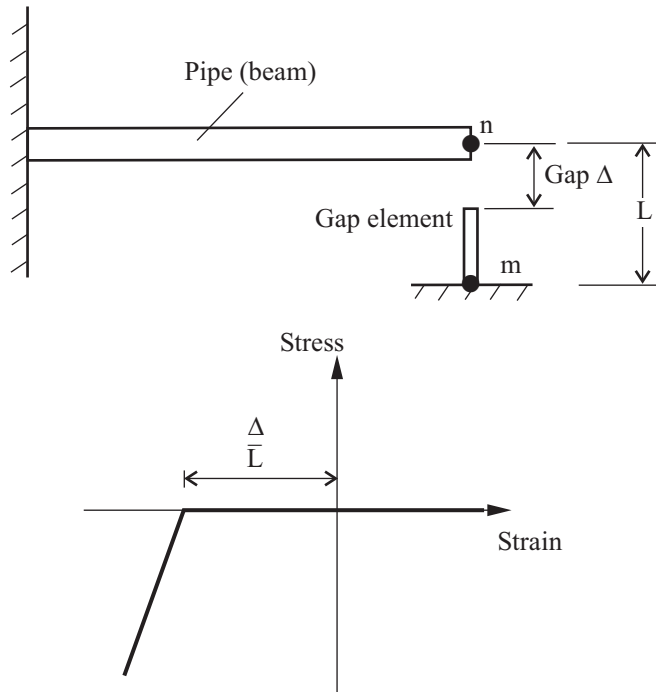


Figure 3.3-4: Modeling of gaps

- The model shown in Fig. 3.3-5(a) corresponds to a compression-only behavior.
- The model shown in Fig. 3.3-5(b) corresponds to a tension cut-off behavior, the snapping of a cable, for example.
- The model shown in Fig. 3.3-5(c) corresponds to a behavior exhibiting both tension and compression cut-offs.

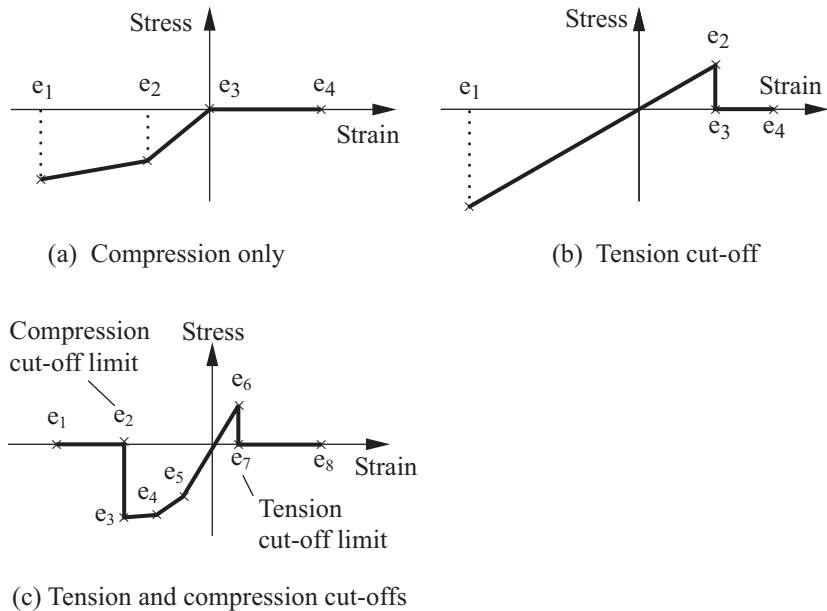


Figure 3.3-5: Various modeling features available with the nonlinear elastic truss model

3.4 Isothermal plasticity material models

ref. KJB
Section 6.6.3

- This section describes the following material models:

Plastic-bilinear, plastic-multilinear: von Mises model with isotropic, kinematic hardening or mixed hardening

Mroz-bilinear: von Mises model with Mroz hardening

Plastic-orthotropic: Hill yielding with bilinear proportional hardening

Gurson: Gurson plastic model for fracture/damage analysis

Plastic-cyclic: von Mises model with hardening rules suitable for modeling cyclic plasticity.

- All elastic-plasticity models use the flow theory to describe the elastic-plastic response; the basic formulations for the von Mises models are summarized on pp. 596-604, ref. KJB, and for the Mroz model in:

ref. Y.F. Dafalias and E.P. Popov, "Plastic Internal Variables Formalism of Cyclic Plasticity," *J. Appl. Mech., Trans. ASME*, Vol. 43, pp. 645-651, 1976.

The formulation for the Ilyushin model is given in the following reference:

ref. K.J. Bathe, E. Dvorkin and L.W. Ho, "Our Discrete-Kirchhoff and Isoparametric Shell Elements – An Assessment," *J. Computers and Structures*, Vol. 16, pp. 89-98, 1983.

- The Drucker-Prager material model is described in Section 3.9.2.

3.4.1 Plastic-bilinear and plastic-multilinear material models

- These material models are based on
 - ▶ The von Mises yield condition (see p. 597, ref. KJB)
 - ▶ An associated flow rule using the von Mises yield function
 - ▶ An isotropic or kinematic, bilinear or multilinear, hardening rule

Figs. 3.4-1 to 3.4-3 summarize some important features of these material models.

- These models can be used with the **truss**, **2-D solid**, **3-D solid**, **beam** (plastic-bilinear only), **iso-beam**, **shell** and **pipe** elements.

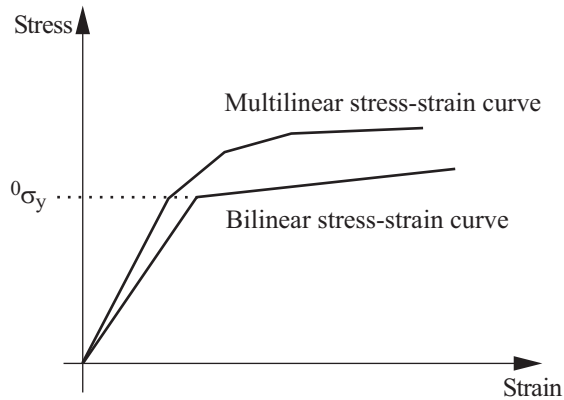


Figure 3.4-1: von Mises model

- These models can be used with the **small displacement/small strain**, **large displacement/small strain** and **large displacement/large strain** formulations. The large displacement/large strain formulation can only be used with the 2-D solid, 3-D solid and shell elements (the shell elements must be 3-node, 4-node, 6-node, 9-node or 16-node single layer shell elements described entirely by midsurface nodes).

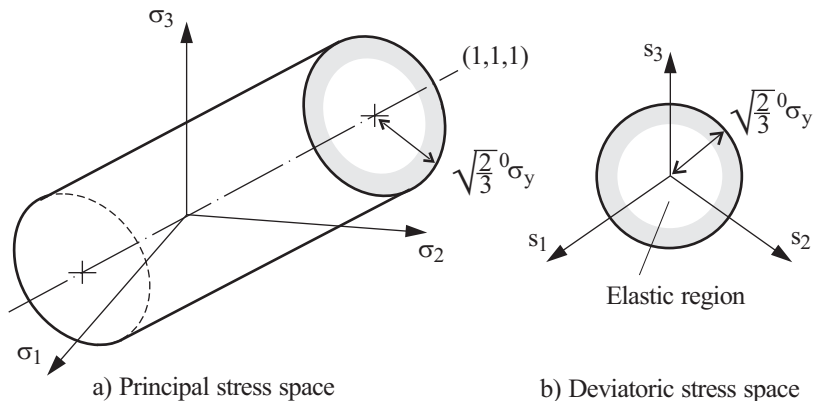


Figure 3.4-2: von Mises yield surface

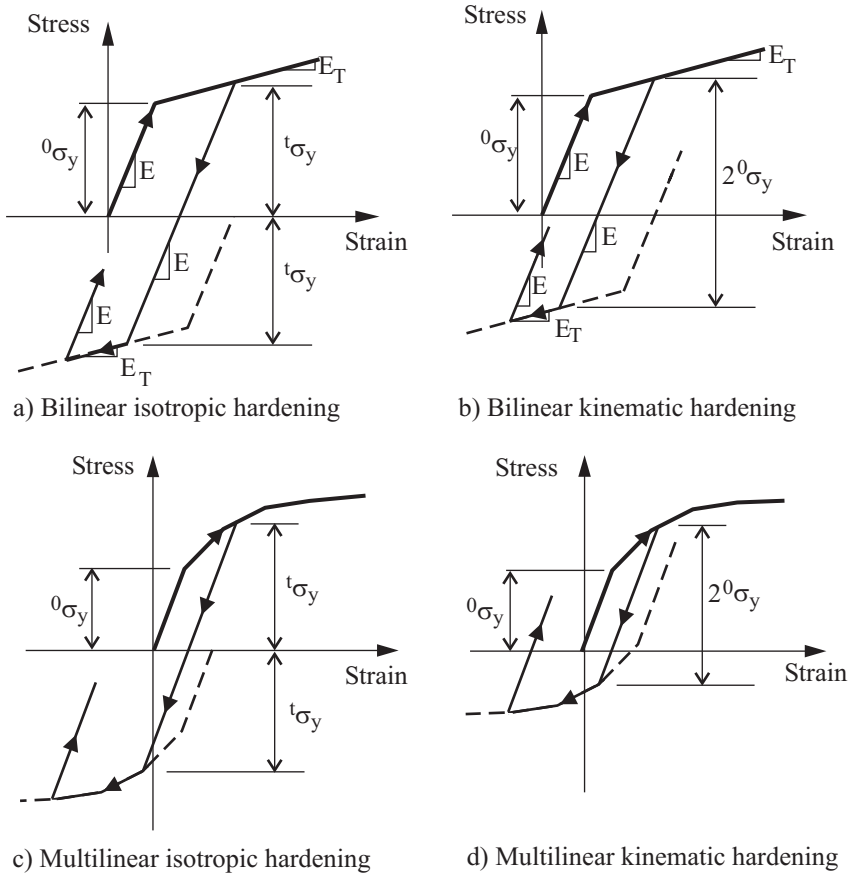


Figure 3.4-3: Isotropic and kinematic hardening

When used with the small displacement/small strain formulation, a materially-nonlinear-only formulation is employed, when used with the large displacement/small strain formulation, either a TL or a UL formulation is employed (depending on element type), and when used with the large displacement/large strain formulation, either a ULH or ULJ formulation is employed.

- If geometrically nonlinear effects are to be included, the large displacements/large strain kinematics are preferred to the large displacement/small strain kinematics, even when the strains are numerically small. The large displacement/small strain kinematics should be used only when the large displacement/large strain

kinematics are not supported by the element. By default large strain kinematics are used for 2-D solid and 3-D solid elements when large displacement kinematics are selected.

- For multilinear plasticity, there is no restriction on the number of stress-strain points in the stress-strain curve.
- Mixed hardening is available only in bilinear plasticity.
- Plane strain, axisymmetric or 3-D solid elements that reference these material models should also employ the mixed displacement-pressure (u/p) element formulation. The u/p formulation is used by default for these elements.
- In the von Mises model with isotropic hardening, the following yield surface equation is used:

$${}^t f_y = \frac{1}{2} ({}^t \mathbf{s} \cdot {}^t \mathbf{s}) - \frac{1}{3} {}^t \sigma_y^2 = 0$$

where ${}^t \mathbf{s}$ is the deviatoric stress tensor and ${}^t \sigma_y$ the updated yield stress at time t .

In the von Mises model with kinematic hardening, the following yield surface equation is used:

$${}^t f_y = \frac{1}{2} ({}^t \mathbf{s} - {}^t \boldsymbol{\alpha}) \cdot ({}^t \mathbf{s} - {}^t \boldsymbol{\alpha}) - \frac{1}{3} {}^0 \sigma_y^2 = 0$$

where ${}^t \boldsymbol{\alpha}$ is the shift of the center of the yield surface (back stress tensor) and ${}^0 \sigma_y^2$ is the virgin, or initial, yield stress.

In the von Mises model with mixed hardening, the following yield surface equation is used:

$${}^t f_y = \frac{1}{2} ({}^t \mathbf{s} - {}^t \boldsymbol{\alpha}) \cdot ({}^t \mathbf{s} - {}^t \boldsymbol{\alpha}) - \frac{1}{3} {}^t \sigma_y^2 = 0$$

where

$${}^t\sigma_y = {}^0\sigma_y + ME_p \mathbf{e}^p$$

The back stress ${}^t\boldsymbol{\alpha}$ is evolved by

$$d\boldsymbol{\alpha} = C_p (1 - M) d\mathbf{e}^p$$

C_p is Prager's hardening parameter, related to the plastic modulus E_p by

$$C_p = \frac{2}{3} E_p$$

and M is the factor used in general mixed hardening ($0 < M < 1$) which can be a variable, expressed as

$$M = M_\infty + (M_0 - M_\infty) \exp(-\eta e^p)$$

The formulation for the von Mises model with mixed hardening is given in the following reference:

ref K.J. Bathe and F.J. Montáns, "On Modeling Mixed Hardening in Computational Plasticity", *Computers and Structures*, Vol. 82, No. 6, pp. 535 - 539, 2004.

Note that the convergence might not be good when the mixed hardening parameter $\eta \neq 0$. Therefore, $\eta = 0$ is preferred.

The yield stress is a function of the effective plastic strain, which defines the hardening of the material. The effective plastic strain is defined as

$${}^t e^{-p} = \int_0^t \sqrt{\frac{2}{3}} d\mathbf{e}^p \cdot d\mathbf{e}^p$$

in which $d\mathbf{e}^p$ is the tensor of differential plastic strain increments and in which $d\mathbf{e}^p \cdot d\mathbf{e}^p$ is calculated as $de_{ij}^p de_{ij}^p$ (see ref. KJB, p.

599). In finite element analysis, we approximate ${}^t e^{-p}$ as the sum of all of the plastic strain increments up to the current solution time:

$${}^t\bar{e}^{-P} = \sum_{\text{all solution steps}} \Delta e^{-P}$$

where $\Delta e^{-P} = \sqrt{\frac{2}{3} \Delta \mathbf{e}^p \cdot \Delta \mathbf{e}^p}$ and $\Delta \mathbf{e}^p$ is the tensor of plastic strain increments in a solution step. Because of the summation over the solution steps, we refer to the calculated value of ${}^t\bar{e}^{-P}$ as the accumulated effective plastic strain.

- If a thermal load is applied to the structure, the thermal strains are taken into account but the material characteristics are considered to be temperature independent.

Stress-strain input data for multilinear plasticity in large strain analysis

In large strain analysis, ADINA works internally with true (Cauchy) stresses and true (logarithmic) strains. However, typical tension tests used to determine experimental data return forces and displacements. These forces and displacements are used to compute engineering stresses (force per unit original area) and engineering strain (change in length per unit length). Therefore it is necessary to convert engineering stress-strain data to true stress-strain data. This conversion is either done by the user, or is done automatically by the AUI (using the MASTER CONVERT-SSVAL=YES option). We now discuss this conversion process in detail.

Consider an experiment in which a fully incompressible material is put into uniaxial tension. The force-displacement curve is determined, and the following information extracted from the force-displacement curve:

The engineering stress is computed as the force / original area.
The engineering strain is computed as the displacement / original length.

This data is determined for a number of points on the force-displacement curve, starting at point 1, which is considered to be the elastic limit. And the Young's modulus is also determined as the ratio of engineering stress / engineering strain at the elastic limit.

Now consider duplicating the experimental results with a finite element model that uses a large strain formulation. The tension test finite element model should return the correct force for each displacement point on the force-displacement curve.

Because the stress and strain measures used in the large strain finite element formulation are true stress and true strain, it is necessary to convert the engineering stress / engineering strain data to true stress / true strain data.

The conversion can be done using an algorithm similar to

```
{
  For (each stress-strain point i, i=1, 2, ... ) {
     $e_i$  = engineering strain
     $\sigma_i$  = engineering stress
     $\tau_i = \sigma_i(1 + e_i)$  = true stress
     $\varepsilon_i = \ln(1 + e_i)$  = true strain
  }
   $E = \tau_1 / \varepsilon_1$ 
}
```

Notice that E also needs to be converted. The reason is as follows. If E is not converted, ADINA computes $\varepsilon_1 = \tau_1 / E$ but now ε_1 is no longer the true strain at the elastic limit.

Here is a numerical example:

$$e_1 = 0.1, \sigma_1 = 30 \text{ MPa}, E = \sigma_1 / e_1 = 300 \text{ MPa}$$

The conversion given above produces

$$\varepsilon_1 = 0.09531, \tau_1 = 33 \text{ MPa}$$

This point is assumed to be at the elastic limit. If E is kept at 300, then ADINA computes

$$\varepsilon_1 = 33 / 300 = 0.11$$

and this ε_1 is no longer at the elastic limit. Therefore E must be

recalculated as part of the conversion process:

$$E = 33 / 0.09531 = 346.2 \text{ MPa}$$

Assuming that the above conversion is performed, then the tension test finite element model will return the correct force for each of the displacements corresponding to the points on the original force-displacement curve.

Assumptions in the conversion formulas

There are a number of assumptions in the above formulas, as follows:

Elastic response

It is assumed that the stress-strain behavior is linearly elastic up to the elastic limit. However, since ADINA uses true stress / true strain data internally, the linear elastic behavior is also based on true stress / true strain data. In the above example, the linear elastic response computed by the tension test finite element model is based on a Young's modulus of 346.2 MPa. This response is, of course, quite different than a linear elastic response based on a Young's modulus of 300 MPa.

The reason that the Young's modulus is so different is because the strain at the elastic limit is "large". When the strain at the elastic limit is small, then the change in Young's modulus is also small. The change in Young's modulus caused by the conversion is approximately equal to $\Delta E = \frac{3}{2} e_1 E$.

Compressible elastic material

The above conversion assumes that the material response is fully incompressible, under both elastic and plastic conditions. However, in most cases, the material is compressible under elastic conditions.

Nevertheless, the above formulas are frequently used anyway. The error thus incurred will be largest for point 1, and will diminish for larger values of strain.

Response in uniaxial compression

The above formulas use tension data to convert from engineering to true values. However, it should be recognized that ADINA uses the same stress-strain curve in both tension and compression. Therefore, if a large strain finite element model is put into uniaxial compression, the observed (compressive) force will not be equal in magnitude to the expected (tensile) force, at a given (compressive) displacement level.

Here is an example. Suppose that in a tensile test, the following data is obtained:

Point	Engineering strain	Engineering stress (MPa)	True strain	True stress
...				
3	0.05	50	0.04879	52.5
4	0.05263	51	0.05129	53.7
...				

First consider a materially-nonlinear-only analysis. Enter the engineering stress and strain values for the two points as part of the stress-strain input data. When the model is run in compression to an engineering strain of -0.05, the engineering stress is -50 MPa. The force-displacement response is symmetric in tension and compression.

Next consider a large strain analysis. Enter the true stress and strain values for the two points as part of the stress-strain input data. Now run the model in compression to an engineering strain of -0.05. An engineering strain of -0.05 corresponds to a true strain of -0.05129, therefore the true stress is -53.7 and the engineering stress is -56.5. The force-displacement response is not symmetric in tension and compression.

Homogeneous deformation

The above conversion assumes that the specimen is uniform and that the specimen deforms homogeneously under load. Therefore the conversion is only valid up to the ultimate tensile strength of the material, because beyond that point the deformation of the specimen might be no longer homogeneous due to localized

necking.

MASTER CONVERT-SSVAL feature

When CONVERT-SSVAL=NO in the MASTER command (the default), the AUI does not perform any conversion. True stress-strain data should be input, and the user could compute this true stress-strain data using the formulas given above.

When CONVERT-SSVAL=YES, then the AUI performs this conversion using the formulas given above. Therefore engineering stress-strain data should be input.

The CONVERT-SSVAL=YES feature should only be used when all of the elements that use multilinear plastic materials also use large strain formulations.

The CONVERT-SSVAL=YES feature does not perform any conversions for the bilinear plastic material models, or for the plastic-cyclic material models.

Material behavior beyond the last point of the stress-strain curve in multilinear plasticity

The material behavior beyond the last point of the stress-strain curve in multilinear plasticity can be considered ruptured, or the curve can be extended indefinitely with the slope of its final segment. This depends on the global setting of the EXTEND-SSCURVE paramter in the MASTER command with indefinite extension as the default.

Modeling of rupture: Rupture conditions can also be modeled with these material models except for the beam element. For the bilinear stress-strain curve, a maximum allowable effective plastic strain $\bar{\epsilon}_A^P$ can be specified for the rupture condition. For the multilinear stress-strain curve, the rupture plastic strain corresponds to the effective plastic strain at the last point input for the stress-strain curve.

When rupture is reached at a given element integration point, the corresponding element is removed from the model (see Section 11.5).

There is also the option of user-supplied rupture. You code the rupture condition into one of the CURUP subroutines: CURUP2 for 2-D solid elements, CURUP3 for 3-D solid elements, CURUP4

for beam elements, CURUP5 for iso-beam elements, CURUP7 for shell elements and CURUP8 for pipe elements. These subroutines are in file ovlusr.f. ADINA provides the calculated latest stresses, total strains, thermal strains, plastic strains, creep strains, yield stress and accumulated effective plastic strain to these subroutines and these subroutines return the current rupture state.

In subroutines CURUP2, CURUP3 and CURUP7, ADINA also provides solution time, accumulated effective creep_strain (AECST2) of the creep material model, and user-input real and integer arrays RRUPTR and IRUPTR. Viscoplastic strains and accumulated effective viscoplastic strains of the Anand material model are also provided through EPSP2 and EPSTR2.

Rate-dependent plasticity for truss, 2-D solid, 3-D solid, isobeam, pipe and shell elements: The rate dependent model in ADINA is used to simulate the increase in the yield stress with an increase in strain rate.

The rate-dependent model only applies to the isotropic plasticity models with isotropic hardening (bilinear or multilinear).

The rate-dependent model is implemented for truss, 2-D solid, 3-D solid, isobeam, pipe and shell (either single-layered or multilayered) elements.

The effective yield stress including strain rate effects is

$$\sigma_y = \sigma_y^0 \left[1 + b \ln \left(1 + \frac{\dot{\epsilon}^P}{\dot{\epsilon}_0} \right) \right]$$

where σ_y^0 is the static yield stress, $\dot{\epsilon}_0$ is the transition strain rate and b is the strain rate hardening parameter. Here $\dot{\epsilon}^P$ is calculated using $\dot{\epsilon}^P = \frac{\Delta \bar{\epsilon}^P}{\Delta t}$.

For more information on this formula, see the following reference:

- ref. W.H. Drysdale and A.R. Zak, “Mechanics of Materials and Structural Theories — A Theory for Rate Dependent Plasticity”, *Computers and Structures*, Vol. 20, pp. 259-264, 1985.

The AUI can determine constants $\dot{\epsilon}_0$ and b using curve-fitting. The input for the curve-fitting is one or more plastic strain rates and associated user-input stress-strain curves. For each plastic strain rate $\dot{\epsilon}^P$, the average overstress ratio $\frac{\sigma_y}{\sigma_y^0}$ is determined from the associated user-input stress-strain curve; then the curve-fitting is performed using the strain rates and associated overstress ratios. If only one strain rate is specified, then $\dot{\epsilon}_0$ must be specified and b is determined using curve-fitting. If two or more strain rates are specified, both $\dot{\epsilon}_0$ and b are determined.

Modeling of gaps using the truss element: When used in conjunction with the 2-node truss element the elastic-plastic material models can be used to model gaps in the elements (the nonlinear elastic model can also be used to model gaps with the truss elements; see Section 3.3.1). In this case the gap elements can only resist compressive loads, i.e., gap elements have no tensile stiffness. The gap width input for each element is used to determine a strain e_{gap} (compressive strain is defined as negative). The compressive stiffness of a gap element is zero if the strain in the element is greater than or equal to e_{gap} and is nonzero when the strain is less than e_{gap} .

3.4.2 Mroz-bilinear material model

- This material model is based on
 - ▶ Isotropic elasticity
 - ▶ An associated flow rule
 - ▶ The Mroz kinematic hardening rule for the yield surface
 - ▶ The Prager kinematic hardening rule for the bounding surface
- The model is described in the following references:
 - ref. Z. Mróz, “On the description of anisotropic work hardening”, *J. Mech. Phys. Solids*, 15 (1967) 163-175.
 - ref. F.J. Montáns, “Implicit algorithms for multilayer J2-

plasticity”, Comput. Meth. Appl. Engrg., 189 (2000) 673-700.

- The Mroz-bilinear model can be used with the **2-D solid** and **3-D solid** elements.
- The Mroz-bilinear model can be used with the **small displacement/small strain, large displacement/small strain** and **large displacement/large strain** formulations.

When used with the small displacement/small strain formulation, a materially-nonlinear-only formulation is employed, when used with the large displacement/small strain formulation, the TL formulation is used, and when the large displacement/large strain formulation is used, the ULH formulation is used.

- Plane strain, axisymmetric or 3-D solid elements that reference this material model should employ the mixed displacement-pressure (u/p) formulation.

- The material behavior is characterized by a uniaxial bilinear stress-strain curve plus a bounding line. A representation of the stress-strain curve is shown in Fig. 3.4-4. Under uniaxial loading, plastic deformation initiates at $\sigma = \sigma_y$, and continues along the line AB until the bounding stress σ_{yB} is reached. With further plastic deformation, the stress follows the bounding line along BC. The material response along ABC is the same as in the case of the plastic-bilinear model with isotropic or kinematic hardening. If the tangent modulus E_{TB} is equal to zero, the deformation along BC would correspond to perfect plasticity.

The uniaxial loading is represented in the principal deviatoric stress space in Fig. 3.4-5. The stress path A' B' C' corresponds to the stress-strain path ABC. When the yield surface has translated so that it comes in contact with the bounding surface at point B' (B), the two surfaces translate together along their common contact point normal along B' C' (BC).

- When the loading is reversed, the material undergoes elastic unloading along CD, and deforms plastically once D is reached, see Fig. 3.4-4. From this point, the material response differs from that of the traditional plastic-bilinear model with kinematic or isotropic

hardening. For example, in the case of kinematic hardening, continued plastic deformation follows $D\bar{F}$ with slope E_{TB} , whereas in the case of the Mroz-bilinear model, continued plastic deformation follows DF with slope E_T . In Fig. 3.4-5, this corresponds to translation of the yield surface along $D'F'$. During this translation, the bounding surface does not translate. When the yield surface again comes in contact with the bounding surface at point F' (F), the two surfaces translate together along their common normal along $F'G'$ (FG). If the loading is again reversed, the elastic unloading of the material follows GH in the stress-strain curve.

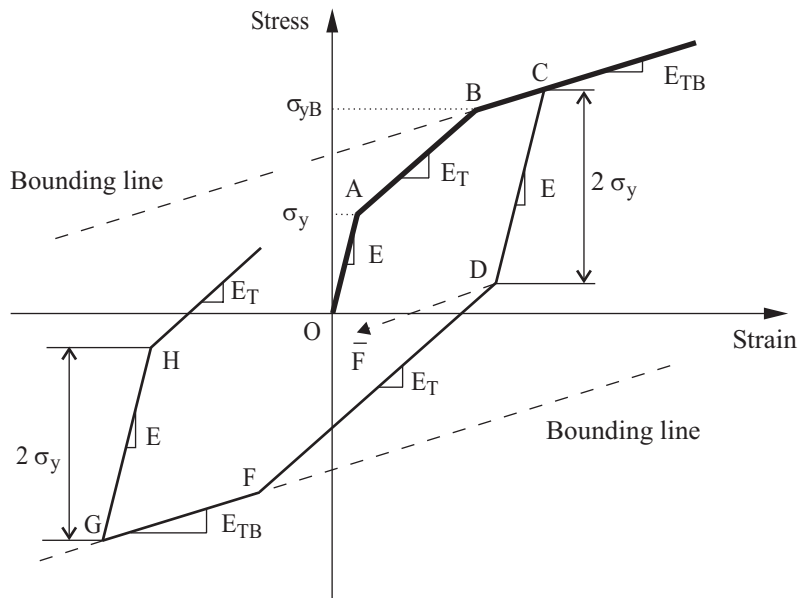


Figure 3.4-4: Mroz-bilinear model; stress-strain curve

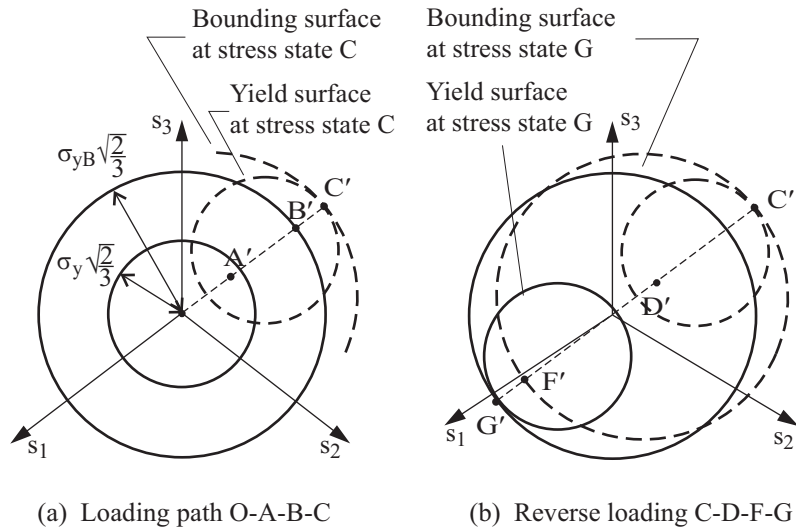


Figure 3.4-5: Mroz-bilinear model behavior in deviatoric stress space

- In order to describe the Mroz-bilinear material model under general, non-radial loading (see Fig. 3.4-6), two conditions are defined for plastic deformation during a time step:
 - ▶ Case I:
 - The yield surface translates in the deviatoric stress space during the time step, but does not come into contact with the bounding surface.
 - ▶ Case II:
 - The yield surface translates in the deviatoric stress space until it reaches the bounding surface, after which both surfaces translate together along their common contact point normal.
 - The two surfaces are already in contact at the beginning of the time step, and the state of the loading causes both surfaces to translate together along their common contact point normal.

The implicit Mroz translation rule for the yield surface is based on a target normal tensor defined as

$${}^{t+\Delta t}\mathbf{t} = \frac{\mathbf{S}^E - {}^t\boldsymbol{\beta}}{\|\mathbf{S}^E - {}^t\boldsymbol{\beta}\|}$$

where \mathbf{S}^E is the trial stress for the time step and ${}^t\boldsymbol{\beta}$ is the back stress of the bounding surface at the end of the previous time step. The target normal tensor is explicit in the sense that no iterations are required to determine it, but implicit in the sense that it depends on the trial stress for the time step. This target normal tensor is the same for both Case I and Case II. In order for the target normal tensor to be uniquely defined, the following condition on the choice of the yield stress and the bounding stress must be satisfied:

$$\sigma_y < \sigma_{yB} < 2\sigma_y$$

Case I:

The translation direction for the yield surface, according to the implicit Mroz translation rule, is given by

$${}^{t+\Delta t}\mathbf{m} = {}^t\boldsymbol{\beta} - {}^t\boldsymbol{\alpha} + \sqrt{\frac{2}{3}} (\sigma_{yB} - \sigma_y) {}^{t+\Delta t}\mathbf{t}$$

with the increment in the back stress of the yield surface given by

$$\Delta\boldsymbol{\alpha} = {}^{t+\Delta t}\boldsymbol{\alpha} - {}^t\boldsymbol{\alpha} = \Delta\gamma^\alpha C_p^\alpha \frac{{}^{t+\Delta t}\mathbf{m}}{\|{}^{t+\Delta t}\mathbf{m}\|}.$$

In the above equation, the effective plastic modulus C_p^α is given by

$$C_p^\alpha = \frac{2}{3} \frac{E E_T}{E - E_T}.$$

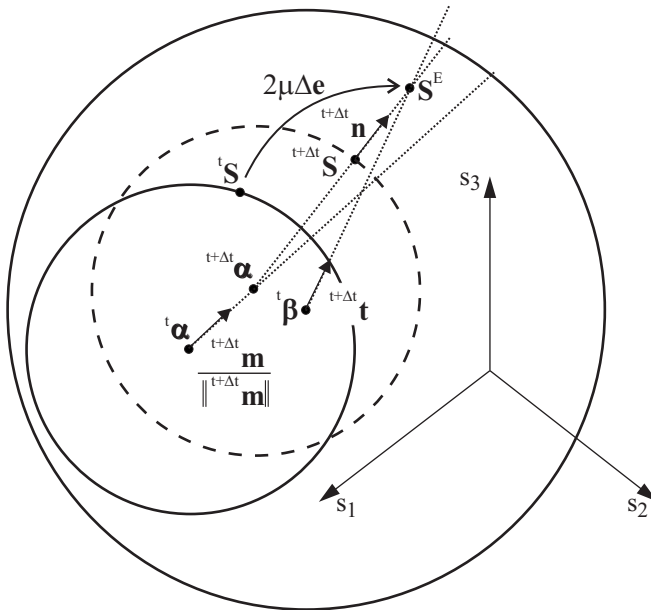
The plastic strain increment for the step is given by the associated

flow rule

$$\Delta \mathbf{e}^p = \Delta \gamma^\alpha {}^{t+\Delta t} \mathbf{n} = \Delta \gamma^\alpha \frac{{}^{t+\Delta t} \mathbf{S} - {}^{t+\Delta t} \boldsymbol{\alpha}}{\| {}^{t+\Delta t} \mathbf{S} - {}^{t+\Delta t} \boldsymbol{\alpha} \|}$$

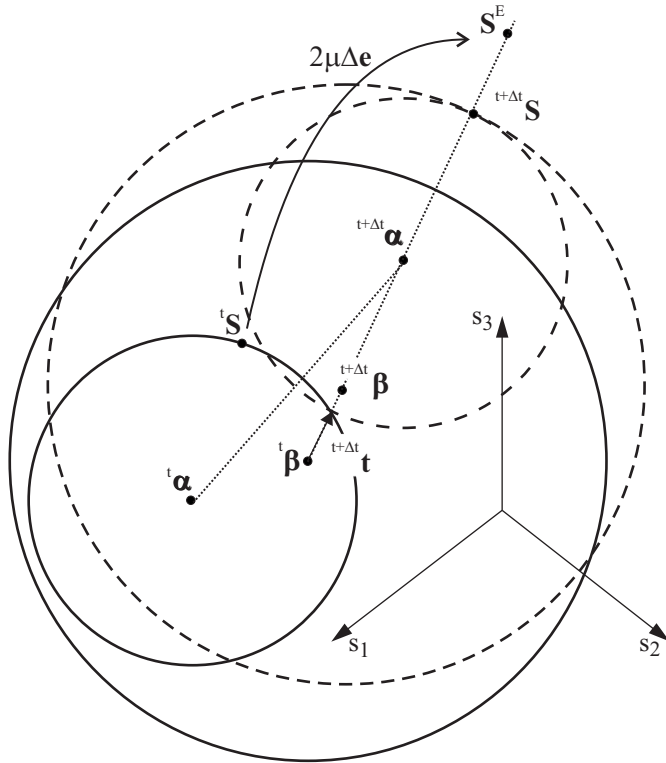
where $\Delta \gamma^\alpha$ is a consistency parameter which has to be determined iteratively. The yield condition $f_y = 0$ is solved for the consistency parameter $\Delta \gamma^\alpha$:

$$f_y = \frac{1}{2} \left({}^{t+\Delta t} \mathbf{S} - {}^{t+\Delta t} \boldsymbol{\alpha} \right) \cdot \left({}^{t+\Delta t} \mathbf{S} - {}^{t+\Delta t} \boldsymbol{\alpha} \right) - \frac{1}{3} \sigma_y^2 = 0.$$



(a) Case I; Only the yield surface translates during the time step

Figure 3.4-6: Mroz model under general, non-radial loading



(b) Case II; Both surfaces translate during the time step

Figure 3.4-6: (continued)

Case II:

The back stress of the bounding surface evolves according to

$$\Delta \boldsymbol{\beta} = {}^{t+\Delta t} \boldsymbol{\beta} - {}^t \boldsymbol{\beta} = \Delta \gamma^\beta C_p^\beta {}^{t+\Delta t} \mathbf{t}$$

where $\Delta \gamma^\beta$ is a consistency parameter associated with translation of the bounding surface, and C_p^β is an effective plastic modulus given by

$$C_p^\beta = \frac{2}{3} \frac{E_T E_{TB}}{E_T - E_{TB}}.$$

By geometry, the position of the yield surface at the end of the time step is related to that of the bounding surface through

$${}^{t+\Delta t}\mathbf{a} = {}^{t+\Delta t}\mathbf{\beta} + \sqrt{\frac{2}{3}}(\sigma_{yB} - \sigma_y) {}^{t+\Delta t}\mathbf{t}.$$

The increment in the back stress of the yield surface becomes

$$\Delta\mathbf{a} = {}^{t+\Delta t}\mathbf{a} - {}^t\mathbf{a} = {}^t\mathbf{\beta} - {}^t\mathbf{a} + \left(\Delta\gamma^\beta C_p^\beta + \sqrt{\frac{2}{3}}(\sigma_{yB} - \sigma_y) \right) {}^{t+\Delta t}\mathbf{t}$$

from which the consistency parameter associated with translation of the yield surface is calculated as

$$\Delta\gamma^\alpha = \frac{\|\Delta\mathbf{a}\|}{C_p^\alpha}.$$

Finally, the plastic strain increment for the step is given by the associated flow rule

$$\Delta\mathbf{e}^p = (\Delta\gamma^\alpha + \Delta\gamma^\beta) {}^{t+\Delta t}\mathbf{t}.$$

As the consistency parameter $\Delta\gamma^\alpha$ is a function of the consistency parameter $\Delta\gamma^\beta$, the yield condition for the bounding surface, $f_{yB} = 0$, can be solved for the consistency parameter $\Delta\gamma^\beta$ alone:

$$f_{yB} = \frac{1}{2} \left({}^{t+\Delta t}\mathbf{S} - {}^{t+\Delta t}\mathbf{\beta} \right) \cdot \left({}^{t+\Delta t}\mathbf{S} - {}^{t+\Delta t}\mathbf{\beta} \right) - \frac{1}{3} \sigma_{yB}^2 = 0$$

- Note that if a zero tangent modulus is prescribed for the bounding surface, $E_{TB} = 0$, the bounding surface does not translate. In this case, the effective stress remains smaller than or

equal to the bounding stress, ${}^{t+\Delta t}\bar{\sigma} = \sqrt{\frac{3}{2} {}^{t+\Delta t}\mathbf{S} \cdot {}^{t+\Delta t}\mathbf{S}} \leq \sigma_{yB}$.

- If a thermal loading is applied to the structure, thermal strains are taken into account, but the material characteristics are taken to be temperature independent.
- Rupture conditions can be modeled: When a positive value \bar{e}_A^p is input for the maximum allowable effective plastic strain, the program compares the accumulated effective plastic strain, ${}^{t+\Delta t}\bar{e}^p$, with \bar{e}_A^p for each material point used in the analysis and over the whole history of deformation. It is considered that the rupture of a material point occurs when ${}^{t+\Delta t}\bar{e}^p$ is greater than \bar{e}_A^p . When rupture is reached at a material point, the corresponding element is removed from the model (see Section 11.5). The user-supplied rupture option is also available (see Section 3.4.1).

3.4.3 Plastic-orthotropic material model

- The plastic-orthotropic model is based on:
 - ▶ The Hill yield condition
 - ▶ An associated flow rule
 - ▶ A proportional hardening rule

The elastic constants can be either isotropic or orthotropic.

- This model can be used with the **2-D solid**, **3-D solid** and **shell** elements.
- The plastic-orthotropic model can be used with the **small displacement/small strain**, **large displacement/small strain** and **large displacement/large strain** formulations. Shell elements used in conjunction with the large displacement/large strain formulation must be 3-node, 4-node, 6-node, 9-node or 16-node single layer elements described entirely by midsurface nodes.

When used with the small displacement/small strain formulation, a materially-nonlinear-only formulation is employed; when used with the large displacement/small strain formulation, the TL formulation is employed; and when used with the large displacement/large strain formulation, the ULJ formulation is employed.

- If geometrically nonlinear effects are to be included, the large displacements/large strain kinematics are preferred to the large displacement/small strain kinematics, even when the strains are numerically small. The large displacement/small strain kinematics should be used only when the large displacement/large strain kinematics are not supported by the element. By default large strain kinematics are used for 2-D solid and 3-D solid elements when large displacement kinematics are selected.
- The mixed formulation can be used (and is the default) for 2-D axisymmetric, 2-D plane strain and 3-D solid elements. But when the mixed formulation is used, the elastic constants must be isotropic.
- The following is a quick summary of the basic concepts used in the plastic-orthotropic material model. For further information, see the following reference:

ref. M. Kojić and K.J. Bathe, *Inelastic Analysis of Solids and Structures*, Springer-Verlag, 2003.

ref. R. Hill, *The Mathematical Theory of Plasticity*, Oxford University Press, 1998.

The Hill yield condition is given by:

$$F(\sigma_{bb} - \sigma_{cc})^2 + G(\sigma_{cc} - \sigma_{aa})^2 + H(\sigma_{aa} - \sigma_{bb})^2 \\ + 2L\sigma_{ab}^2 + 2M\sigma_{ac}^2 + 2N\sigma_{bc}^2 - 1 = 0$$

where (a, b, c) are the material principal axes, and F, G, H, L, M, N are material constants. These constants are given by

$$F = \frac{1}{2} \left(\frac{1}{Y^2} + \frac{1}{Z^2} - \frac{1}{X^2} \right)$$

$$G = \frac{1}{2} \left(\frac{1}{Z^2} + \frac{1}{X^2} - \frac{1}{Y^2} \right)$$

$$H = \frac{1}{2} \left(\frac{1}{X^2} + \frac{1}{Y^2} - \frac{1}{Z^2} \right)$$

$$L = \frac{1}{2Y_{ab}^2}; \quad M = \frac{1}{2Y_{ac}^2}; \quad N = \frac{1}{2Y_{bc}^2}$$

where X, Y, Z , are the yield stresses in the material directions a, b, c , and Y_{ab}, Y_{ac}, Y_{bc} are the yield stresses for pure shear in the planes (a,b) , (a,c) , and (b,c) (see Figure 3.4-7).

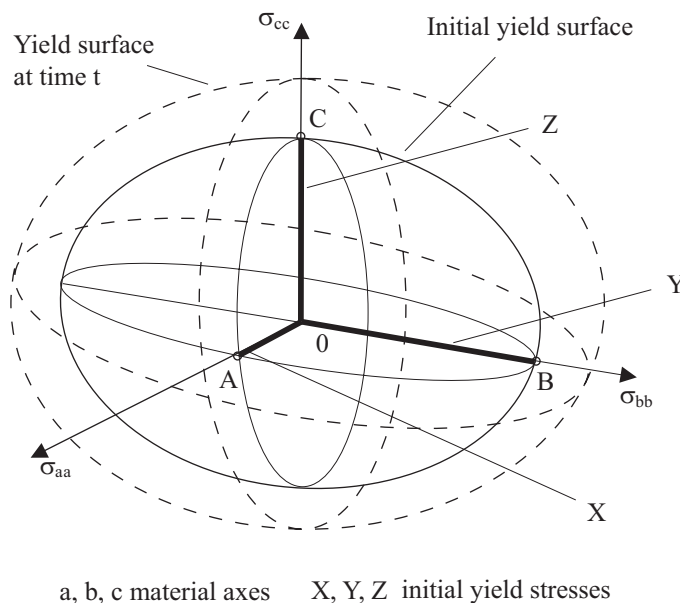


Figure 3.4-7: Orthotropic yield surface

The Hill yield condition can also be written as

$$\begin{aligned} f(\sigma_{bb} - \sigma_{cc})^2 + g(\sigma_{cc} - \sigma_{aa})^2 + h(\sigma_{aa} - \sigma_{bb})^2 \\ + 2\ell\sigma_{ab}^2 + 2m\sigma_{ac}^2 + 2n\sigma_{bc}^2 - \sigma_y^2 = 0 \end{aligned}$$

where

$$f = \sigma_y^2 F, \quad g = \sigma_y^2 G, \quad h = \sigma_y^2 H, \quad \ell = \sigma_y^2 L, \quad m = \sigma_y^2 M, \quad n = \sigma_y^2 N$$

and

$$\sigma_y^2 = \frac{1}{2} \left(\frac{1}{3} (X^2 + Y^2 + Z^2) + Y_{ab}^2 + Y_{ac}^2 + Y_{bc}^2 \right)$$

The Hill effective stress is defined as

$$\begin{aligned} \bar{\sigma}_a^2 = & f(\sigma_{bb} - \sigma_{cc})^2 + g(\sigma_{cc} - \sigma_{aa})^2 + h(\sigma_{aa} - \sigma_{bb})^2 \\ & + 2\ell\sigma_{ab}^2 + 2m\sigma_{ac}^2 + 2n\sigma_{bc}^2 \end{aligned}$$

so that the yield condition can be written $\bar{\sigma}_a^2 - \sigma_y^2 = 0$. Notice that under uniaxial tension, the Hill effective stress is not equal to the uniaxial stress, in general.

The accumulated effective plastic strain \bar{e}_a^P is defined using the principle of the equivalence of plastic work $dW^P = \bar{\sigma}_a d\bar{e}_a^P$. The formula for the increment of effective plastic strain is different than the corresponding formula used for isotropic plasticity. Again, under uniaxial tension, the accumulated effective plastic strain is not equal to the uniaxial plastic strain, in general.

The hardening rule is defined in terms of the relationship between the yield stress and the accumulated effective plastic

strain. The universal plastic modulus is defined as $E_u^P = \frac{d\sigma_y}{d\bar{e}_a^P}$,

and can be either constant or varying, as discussed in more detail below. During hardening, the shape coefficients f, g, h, ℓ, m, n are considered constant.

Note that orthotropic proportional hardening reduces to isotropic hardening for the appropriate values of the elastic and tangent moduli.

The effective-stress-function algorithm is used to calculate stresses and plastic strains when plasticity occurs.

- There are two options for the input of the universal plastic

modulus and the initial yield stress:

Bilinear universal plastic modulus E_u^p . In this case the plastic moduli for directions a, b, c and planes $(a,b), (a,c), (b,c)$ are given by $E_a^p = E_b^p = E_c^p = E_u^p$ and $E_{ab}^p = E_{ac}^p = E_{bc}^p = \frac{E_u^p}{\sqrt{3}}$.

Either enter E_u^p directly, or enter the independent moduli E_a^T, E_b^T, E_c^T for the a, b, c directions and for the $(a,b), (a,c), (b,c)$ planes. Then E_u^p is equal to

$$\sqrt{\frac{1}{2} \left(\frac{1}{3} (E_a^{p^2} + E_b^{p^2} + E_c^{p^2}) + E_{ab}^{p^2} + E_{ac}^{p^2} + E_{bc}^{p^2} \right)}$$

where

$$E_i^p = \frac{E_i E_i^T}{E_i - E_i^T} \quad i = a, b, c$$

and

$$E_{ij}^p = \frac{E_{ij} E_{ij}^T}{E_{ij} - E_{ij}^T} \quad ij = ab, ac, bc$$

The initial yield stress σ_y is entered by direct specification of the yield stresses $X, Y, Z, Y_{ab}, Y_{ac}, Y_{bc}$, then by the formula

$$\sigma_y^2 = \frac{1}{2} \left(\frac{1}{3} (X^2 + Y^2 + Z^2) + Y_{ab}^2 + Y_{ac}^2 + Y_{bc}^2 \right).$$

Analytical uniaxial plastic stress-strain curve: The plastic stress-strain curve is assumed to have the following analytical form:

$$\sigma_y = C \left(e_0 + \bar{e}_a^p \right)^n$$

in which e_0 is the initial yield strain, n is the strain hardening

coefficient and C is a constant. The initial yield stress σ_y is computed using $\sigma_y = Ce_0^n$, and the universal plastic modulus E_u^p is computed using $E_u^p = nC(e_0 + \bar{e}_a^p)^{n-1}$.

- There are three ways to specify material constants f , g , h , ℓ , m , n :

► Enter the yield stresses X , Y , Z , Y_{ab} , Y_{ac} , Y_{bc} then ADINA calculates the material constants using the equations given above. When this option is used in conjunction with an analytical plastic stress-strain curve, the initial yield stress computed from $\sigma_y^2 = \frac{1}{2} \left(\frac{1}{3} (X^2 + Y^2 + Z^2) + Y_{ab}^2 + Y_{bc}^2 + Y_{ac}^2 \right)$ must equal the initial yield stress computed from $\sigma_y = Ce_0^n$.

► Directly specify f , g , h , ℓ , m , n . Note that f , g , h , ℓ , m , n must all be positive. When this option is used in conjunction with a bilinear universal plastic modulus, all of the yield stresses X , Y , Z , Y_{ab} , Y_{ac} , Y_{bc} must be entered, but only the initial yield stress computed from

$$\sigma_y^2 = \frac{1}{2} \left(\frac{1}{3} (X^2 + Y^2 + Z^2) + Y_{ab}^2 + Y_{ac}^2 + Y_{bc}^2 \right) \text{ is used.}$$

- Enter the Lankford coefficients r_0 , r_{45} , r_{90} , then

$$f = \frac{r_0}{r_{90}(r_0 + 1)}, g = \frac{1}{r_0 + 1}, h = \frac{r_0}{r_0 + 1},$$

$$2\ell = \frac{(r_0 + r_{90})(2r_{45} + 1)}{r_{90}(r_0 + 1)}, m = 1.5, n = 1.5$$

in which r_0 is the Lankford coefficient for the rolling direction, r_{45} is the Lankford coefficient for the direction 45° to the rolling direction and r_{90} is the Lankford coefficient for the direction 90° to the rolling direction (the transverse direction). When this

option is used in conjunction with a bilinear universal plastic modulus, all of the yield stresses X , Y , Z , Y_{ab} , Y_{ac} , Y_{bc} must be entered, but only the initial yield stress computed from

$$\sigma_y^2 = \frac{1}{2} \left(\frac{1}{3} (X^2 + Y^2 + Z^2) + Y_{ab}^2 + Y_{ac}^2 + Y_{bc}^2 \right) \text{ is used.}$$

- A typical combination of input options is the analytical universal plastic stress-strain curve option together with Lankford coefficients. When this combination of options is used, the material a -direction is the rolling direction, and, under uniaxial tension in the rolling direction, the Hill effective stress is equal to the uniaxial stress and the effective plastic strain is equal to the uniaxial plastic strain. Therefore the analytical plastic stress-strain constants are taken from the plastic stress-strain curve in the rolling direction. In particular, $\sigma_y = Ce_0^n$ gives the initial yield stress in the rolling direction.
- When the 2-D solid element or shell element is used, material axes a and b must lie in the plane of the element.
- If a thermal load is applied to the structure, the thermal strains are taken into account but the material characteristics are considered to be temperature independent.
- Rupture conditions can also be modeled: a maximum allowable effective plastic strain \bar{e}_a^P can be specified for the rupture condition. When rupture is reached at a given element integration point, the corresponding element is removed from the model (see Section 11.5).
The user-supplied rupture option can also be used (see Section 3.4.1).

3.4.4 Gurson material model

- The Gurson model is intended for use in fracture/damage analysis. The purpose of the Gurson plastic model is to predict ductile crack growth by void growth and coalescence. This is a micromechanical model based approach.

- The Gurson model is available for the **2-D solid** and **3-D solid** elements.

- The Gurson model can be used with the **small displacement/small strain, large displacement/small strain** and **large displacement/large strain** formulations.

When used with the small displacement/small strain formulation, a materially-nonlinear-only formulation is employed, when used with the large displacement/small strain formulation, the TL formulation is employed and when used with the large displacement/large strain formulation, the ULH formulation is employed.

- In the Gurson model, the yield function has the form

$$\phi = \left(\frac{q}{\sigma_0} \right)^2 + 2q_1 f^* \cosh \left(-\frac{3}{2} \frac{q_2 p}{\sigma_0} \right) - (1 + q_3 f^{*2}) = 0$$

in which ϕ is the yield function, q is the von Mises stress, σ_0 is the equivalent tensile flow stress, p is the pressure (positive in compression), f^* is based on the void volume fraction and q_1, q_2, q_3 are the Tvergaard constants. Note that this reduces to the von Mises yield function when f^* equals zero. f^* is defined as

$$f^* = \begin{cases} f & \text{for } f \leq f_c \\ f_c + \frac{f_u - f_c}{f_F - f_c} (f - f_c) & \text{for } f > f_c \end{cases}$$

in which f is the void volume fraction, f_c is the critical void volume fraction, f_F is the void volume fraction at final failure and $f_u^* = 1/q_1$.

- Initially, the void volume fraction is f_0 , which must be input. The increase of void volume fraction is controlled by

$$df = (1 - f) d\varepsilon_{kk}^p + A d\bar{\varepsilon}^p$$

in which $d\varepsilon_{kk}^p$ is the sum of the normal plastic strain components, $\bar{\varepsilon}^p$ is the equivalent plastic strain and A is the void nucleation intensity. A is defined as

$$A = \frac{f_N}{S_N \sqrt{2\pi}} \exp \left[-\frac{1}{2} \left(\frac{\bar{\varepsilon}^p - \varepsilon_N}{S_N} \right)^2 \right]$$

in which f_N is the volume fraction of void nucleating particles, ε_N is the mean void nucleation burst strain and S_N is the corresponding standard deviation.

- The uniaxial stress-strain curve is

$$\frac{\sigma_0}{\sigma_y} = \left(\frac{\sigma_0}{\sigma_y} + \frac{3G}{\sigma_y} \bar{\varepsilon}^p \right)^N$$

where σ_y is the tensile yield stress of the matrix, G is the shear modulus and N is a material constant (typically on the order of 0.1).

- For more information, see the following reference:

ref. N. Aravas, “On the numerical integration of a class of pressure-dependent plasticity models”, *Int. J. Num. Meth. in Engng.*, Vol. 24, 1395-1416 (1987).

3.4.5 Plastic-cyclic material model

- This material model is based on
 - ▶ The von Mises yield condition (see p. 597, ref. KJB)
 - ▶ A flow rule using the von Mises yield function
 - ▶ An isotropic and/or kinematic hardening rule. The isotropic

and kinematic hardening rules used in the plastic-cyclic model are suitable for use in modeling cyclic plasticity.

- The plastic-cyclic material model differs from the plastic-bilinear and plastic-multilinear material models because the isotropic and kinematic hardening rules are different. However, the plastic-cyclic material model can reproduce the plastic-bilinear and plastic-multilinear models when suitable material constants are chosen.
- This material model can be used with the **truss, 2-D solid, 3-D solid, Hermitian beam, shell and 3-D shell** elements. The shell elements must be MITC3, MITC4, MITC6, MITC9 or MITC16 single-layer shell elements.
- This material model can be used with the **small displacement/small strain, large displacement/small strain and large displacement/large strain** formulations. Large displacement/large strain kinematics can only be used with the 2-D solid, 3-D solid, shell and 3D-shell elements.
- If geometrically nonlinear effects are to be included, the large displacements/large strain kinematics are preferred to the large displacement/small strain kinematics, even when the strains are numerically small. The large displacement/small strain kinematics should be used only when the large displacement/large strain kinematics are not supported by the element. By default large strain kinematics are used for 2-D solid and 3-D solid elements when large displacement kinematics are selected.

3.4.5.1 Fundamental concepts

- Many of the ideas used in the plastic-cyclic material model are given in the following reference:

ref J. Lemaitre and J.-L. Chaboche, *Mechanics of Solid Materials*, Cambridge University Press, 1990.

We abbreviate this reference as ref LC in the discussion below. As

an aid to understanding the model, whenever our notation differs from the notation used in ref LC, we give the equivalent Lemaitre and Chaboche notation.

- The motivation for the plastic-cyclic material model is illustrated in Figs. 3.4-8 and 3.4-9. Fig. 3.4-8 shows a bar subjected to uniaxial cycling, with the strain prescribed. Fig. 3.4-9 shows response predictions from the plastic-bilinear and plastic-multilinear material models of Section 3.4.1. On repeated cyclic loading, perfect plasticity and multilinear hardening plasticity produce stabilized plastic cycles with no additional hardening. Bilinear isotropic hardening does not produce a stable plastic cycle and bilinear kinematic hardening produces a very rough approximation to a stable plastic cycle.
- Response predictions from the plastic-cyclic material model are illustrated in Fig. 3.4-10. When nonlinear kinematic hardening is used without isotropic hardening, a stable plastic cycle is reached after one cycle. In this stable plastic cycle, the transition from elastic to plastic conditions occurs more gradually than the corresponding transition from bilinear kinematic hardening. Cyclic hardening and cyclic softening are obtained by combining the nonlinear kinematic hardening with isotropic hardening or softening.

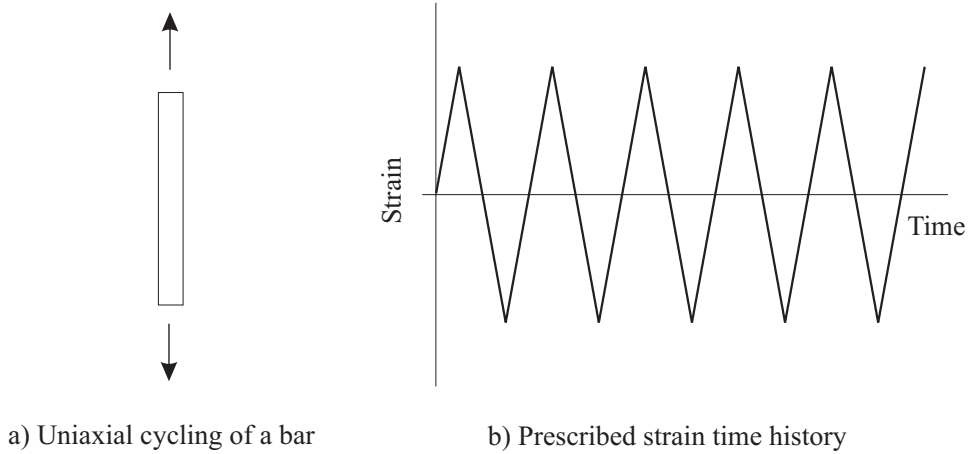


Figure 3.4-8: Uniaxial cycling example

- The plastic-cyclic material model can optionally contain a strain memory surface. The motivation for using the strain memory surface is shown in Fig. 3.4-11. Considering an increase in the prescribed strain amplitude, if no strain memory surface is used, then no additional cyclic hardening takes place, whereas if a strain memory surface is used, additional cyclic hardening takes place.

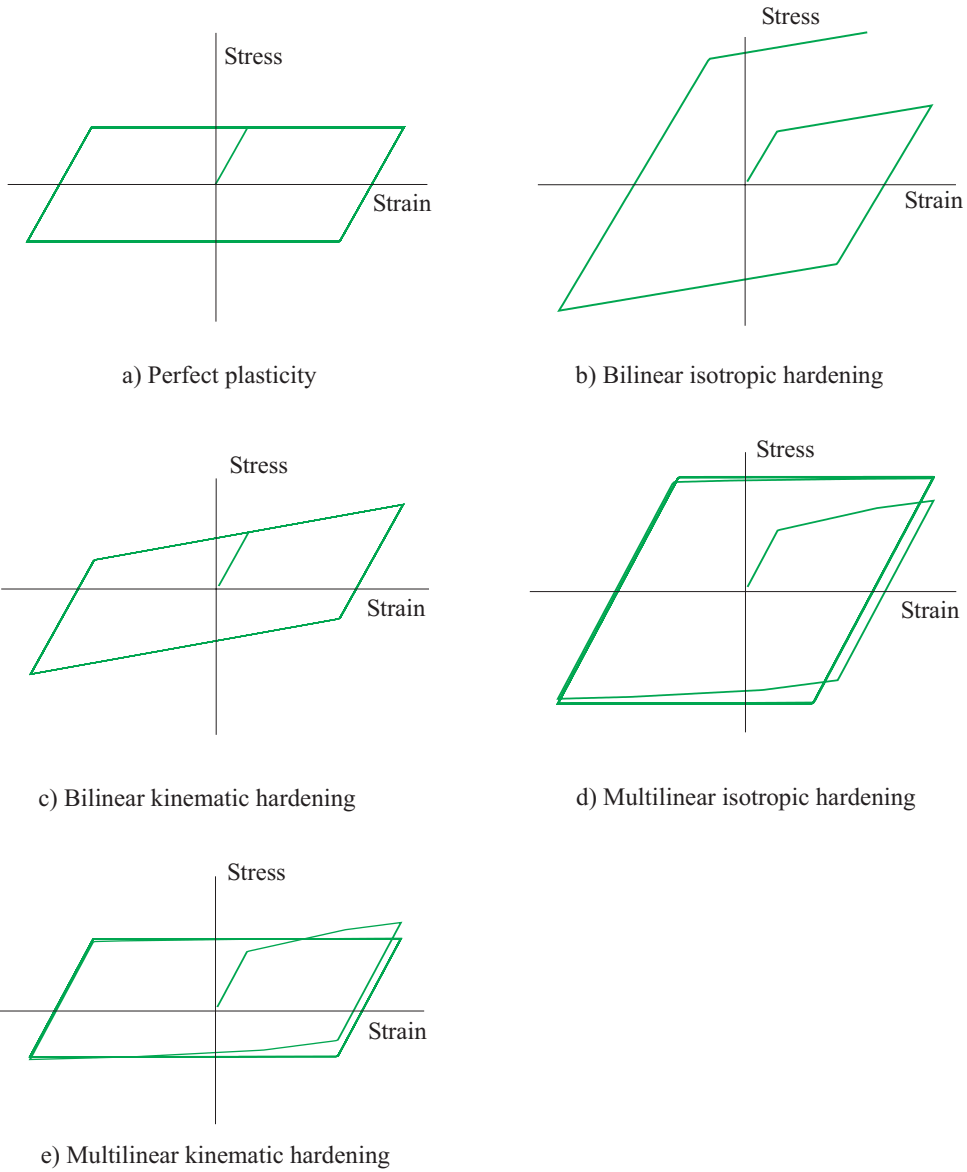
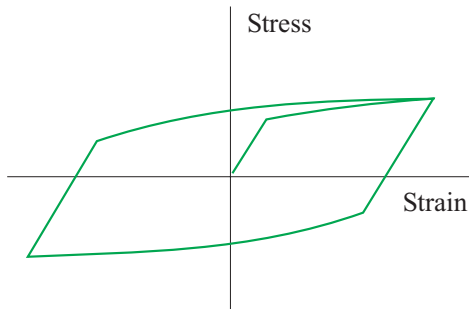
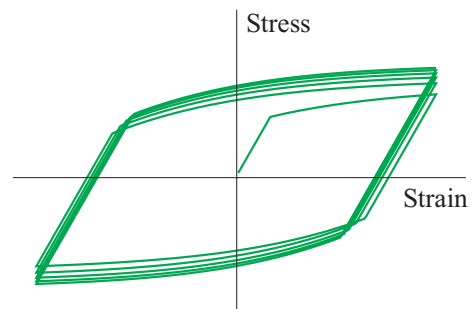


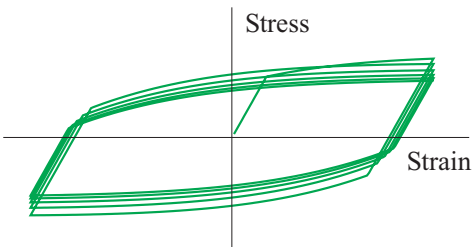
Figure 3.4-9: Response predictions using the plastic-bilinear and plastic-multilinear models



a) Nonlinear kinematic hardening,
no isotropic hardening or softening



b) Nonlinear kinematic hardening,
isotropic hardening



c) Nonlinear kinematic hardening,
isotropic softening

Figure 3.4-10: Response predictions using the plastic-cyclic material model

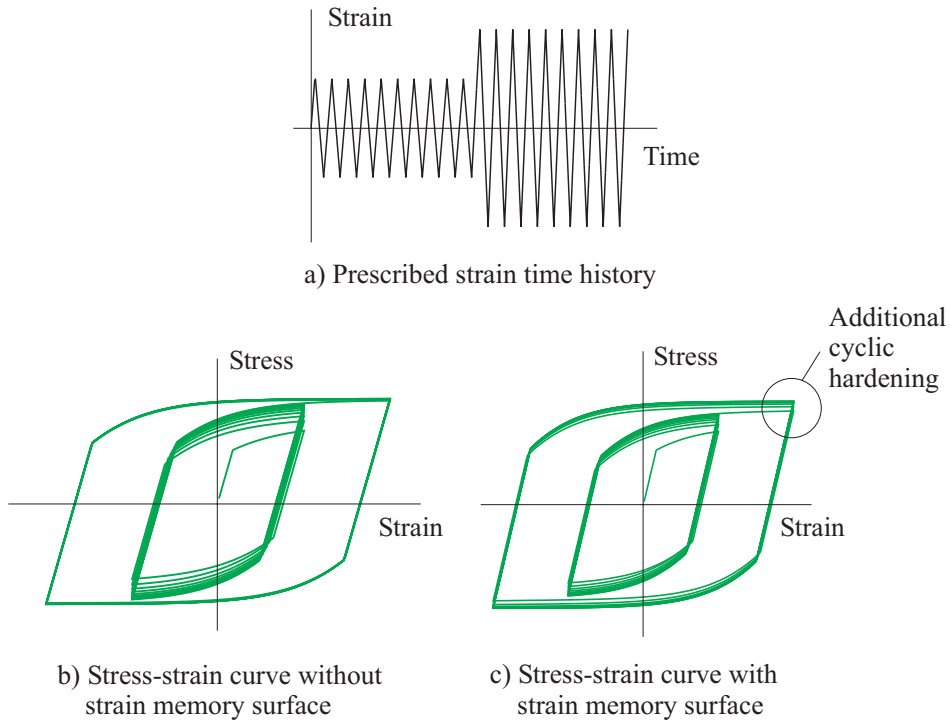


Figure 3.4-11: Response predictions with and without strain memory surface

- Ratchetting occurs when prescribed stresses with non-zero mean stress are applied to the bar, as shown in Fig. 3.4-12.

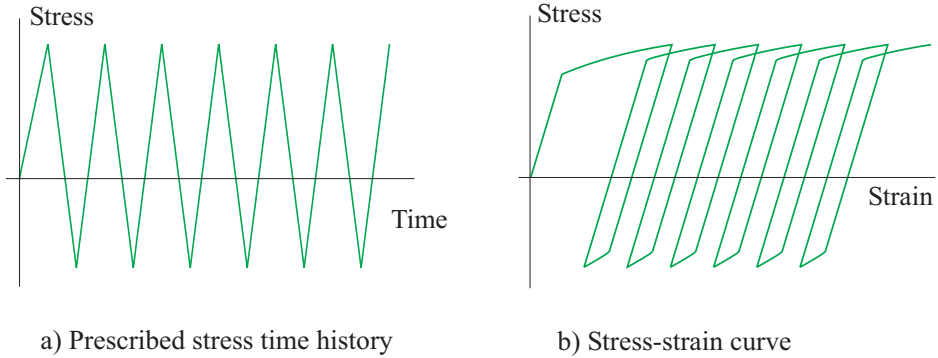


Figure 3.4-12: Ratchetting using the plastic-cyclic material model

Stresses, strains, stress-strain law:

$${}^{t+\Delta t}\boldsymbol{\tau} = {}^{t+\Delta t}\mathbf{s} + {}^{t+\Delta t}\tau_m \mathbf{I}$$

where ${}^{t+\Delta t}\boldsymbol{\tau}$ is the stress tensor ($\boldsymbol{\sigma}$ in ref LC), ${}^{t+\Delta t}\mathbf{s}$ is the deviatoric stress tensor ($\boldsymbol{\sigma}'$ in ref LC) and ${}^{t+\Delta t}\tau_m$ is the mean stress (hydrostatic stress σ_H in ref LC).

$${}^{t+\Delta t}\mathbf{e} = {}^{t+\Delta t}\mathbf{e}' + {}^{t+\Delta t}e_m \mathbf{I}$$

where ${}^{t+\Delta t}\mathbf{e}$ is the strain tensor ($\boldsymbol{\varepsilon}$ in ref LC), ${}^{t+\Delta t}\mathbf{e}'$ is the deviatoric strain tensor ($\boldsymbol{\varepsilon}'$ in ref LC) and ${}^{t+\Delta t}e_m$ is the mean strain (hydrostatic strain ε_H in ref LC).

$$\begin{aligned} {}^{t+\Delta t}\tau_m &= 3\kappa {}^{t+\Delta t}e_m \\ {}^{t+\Delta t}\mathbf{s} &= 2G({}^{t+\Delta t}\mathbf{e}' - {}^{t+\Delta t}\mathbf{e}^P) \end{aligned}$$

where κ and G are the bulk modulus and shear modulus, and ${}^{t+\Delta t}\mathbf{e}^P$ is the plastic strain ($\boldsymbol{\varepsilon}^P$ in ref LC). Thermal strains are not included in the above equations, but are included in ADINA when there are thermal effects.

Yield condition: The von Mises yield condition is

$${}^{t+\Delta t}f_y = \frac{1}{2} \left\| {}^{t+\Delta t}\mathbf{s} - {}^{t+\Delta t}\boldsymbol{\alpha} \right\|^2 - \frac{1}{3} {}^{t+\Delta t}\sigma_y^2 = 0$$

where ${}^{t+\Delta t}\boldsymbol{\alpha}$ is the back stress tensor (\mathbf{X} and \mathbf{X}' in ref LC, note that back stress is always deviatoric) and ${}^{t+\Delta t}\sigma_y$ is the yield stress ($\sigma_Y + R$ or k in ref LC). The norm of a symmetric tensor \mathbf{a} is defined as $\|\mathbf{a}\| = \sqrt{\mathbf{a} : \mathbf{a}}$. The yield condition is always evaluated at time $t + \Delta t$.

Figure 3.4-13 shows the yield condition.

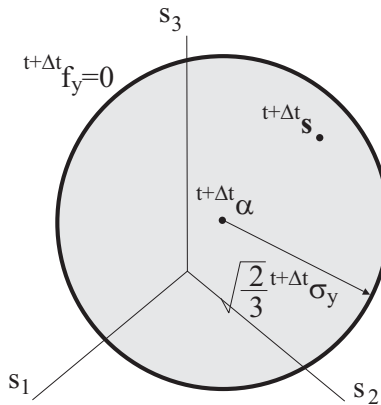


Figure 3.4-13: von Mises yield surface in principal deviatoric stress space

Flow rule: The flow rule states that the direction of plastic strain increments is normal to the yield surface.

$$\text{directions of } \Delta \mathbf{e}^P = \text{directions of } \left({}^{t+\beta\Delta t} \mathbf{s} - {}^{t+\beta\Delta t} \boldsymbol{\alpha} \right)$$

where $\Delta \mathbf{e}^P$ is the increment in plastic strain, and β is a constant used to choose the yield surface configuration ($\beta = 0$ corresponds to the configuration at time t , $\beta = 1$ corresponds to the configuration at time $t + \Delta t$, other values of β correspond to intermediate configurations). (The concept is similar to that used in alpha-integration; we would have used α instead of β , except that α is used for the back stresses.) The directions of a symmetric tensor \mathbf{a} are defined as directions of $\mathbf{a} = \frac{\mathbf{a}}{\|\mathbf{a}\|}$.

Figure 3.4-14 shows the evolution of the yield surface using the stress-strain law, yield condition and flow rule.

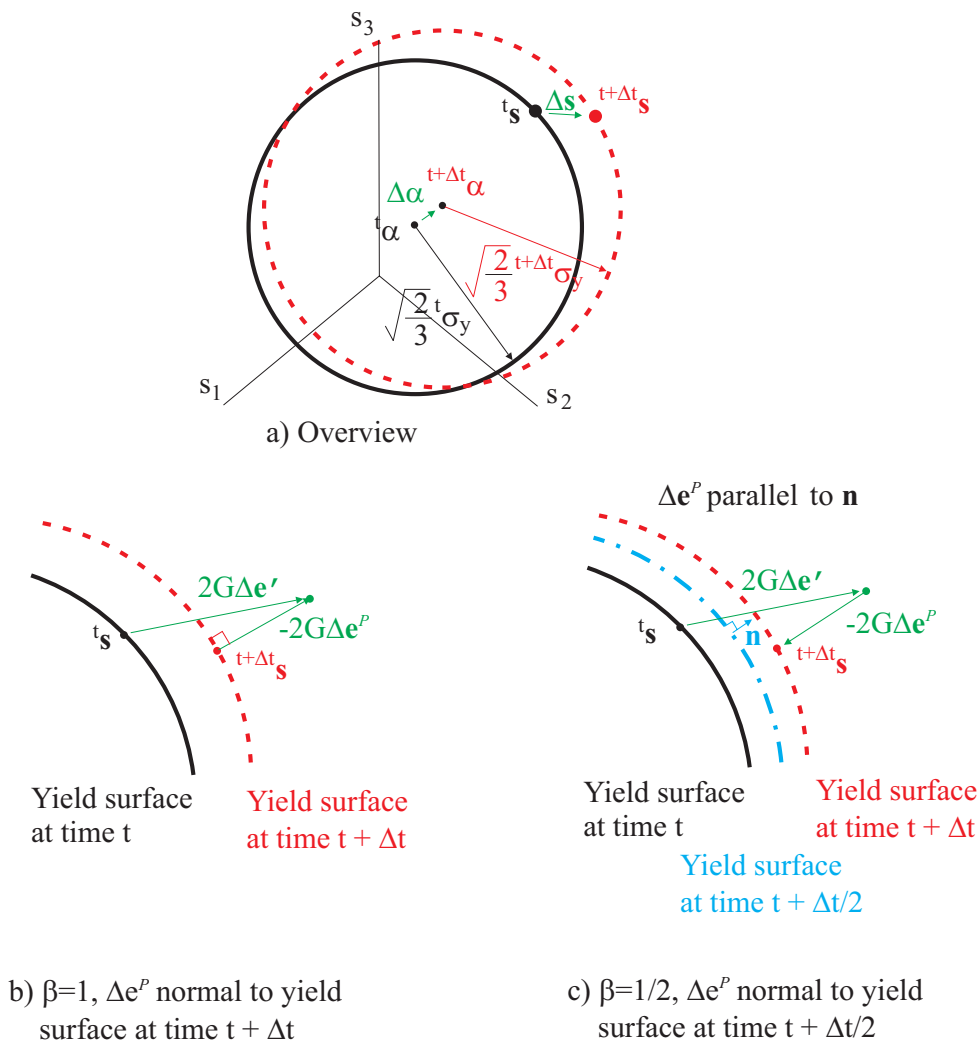


Figure 3.4-14: Incremental update of yield surface and plastic strains

Strain memory surface: The memory-exponential isotropic hardening rule uses the concept of a strain memory surface with additional internal variables ${}^{t+\Delta t}\xi$ and ${}^{t+\Delta t}q$. The strain memory surface is now briefly described.

The strain memory surface is defined in the space of plastic strains as the surface of a sphere centered at position ${}^{t+\Delta t}\xi$ with radius

$\sqrt{\frac{3}{2}} {}^{t+\Delta t}q$. (In ref LC, ${}^{t+\Delta t}\xi$ is written ζ). This surface can be written as

$$f_m = \frac{2}{3} \left\| {}^{t+\Delta t}\mathbf{e}^P - {}^{t+\Delta t}\xi \right\|^2 - {}^{t+\Delta t}q^2 = 0$$

Figure 3.4-15 shows the strain memory surface.

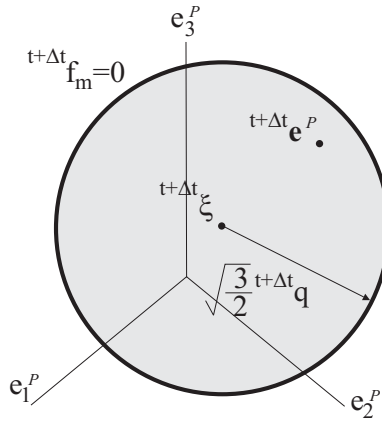


Figure 3.4-15: Strain memory surface in principal plastic strain space

The differential rules used for the evolution of the strain memory surface are

$$dq = \eta \left\langle (\text{directions of } d\mathbf{e}^P) : (\text{directions of } ({}^t\mathbf{e}^P - {}^t\xi)) \right\rangle d\bar{e}^P$$

and

$$\text{directions of } d\xi = \text{directions of } ({}^t\mathbf{e}^P - {}^t\xi)$$

in which the symbol $\langle \rangle$ means that $\langle u \rangle = 0$ when $u < 0$, and $\langle u \rangle = u$ when $u > 0$. The incremental versions of these rules are

$$\Delta q = \eta \left\langle (\text{directions of } \Delta \mathbf{e}^P) : (\text{directions of } ({}^{t+\beta\Delta t} \mathbf{e}^P - {}^{t+\beta\Delta t} \xi)) \right\rangle \Delta \bar{e}^P$$

and

$$\text{directions of } \Delta \xi = \text{directions of } ({}^{t+\beta\Delta t} \mathbf{e}^P - {}^{t+\beta\Delta t} \xi)$$

Again, the β notation is used to denote a configuration between t and $t + \Delta t$. These concepts are illustrated in Fig. 3.4-16.

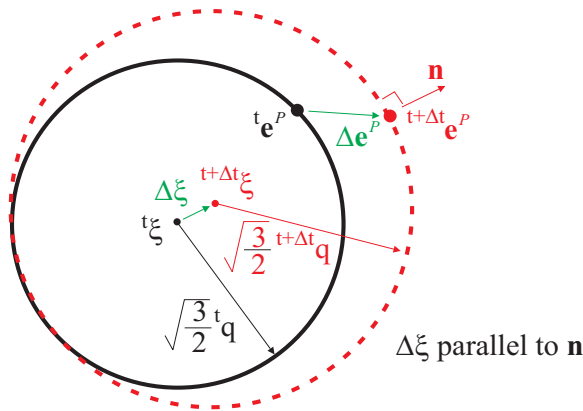


Figure 3.4-16: Incremental update of strain memory surface, for $\beta=1$

η is a material constant that must be between 0 and $\frac{1}{2}$. Typically

$$\eta = \frac{1}{2}.$$

These rules ensure that the strain memory surface at time $t + \Delta t$ encloses the strain memory surface for all preceding times. The evolution of the strain memory surface in 1D uniaxial straining is shown in Figure 3.4-17.

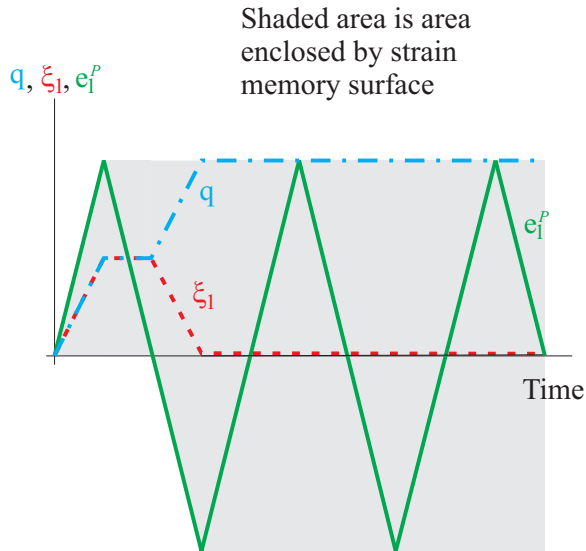


Figure 3.4-17: Evolution of strain memory surface in uniaxial cyclic straining

Isotropic hardening rules: The isotropic hardening rules are shown in Figure 3.4-18.

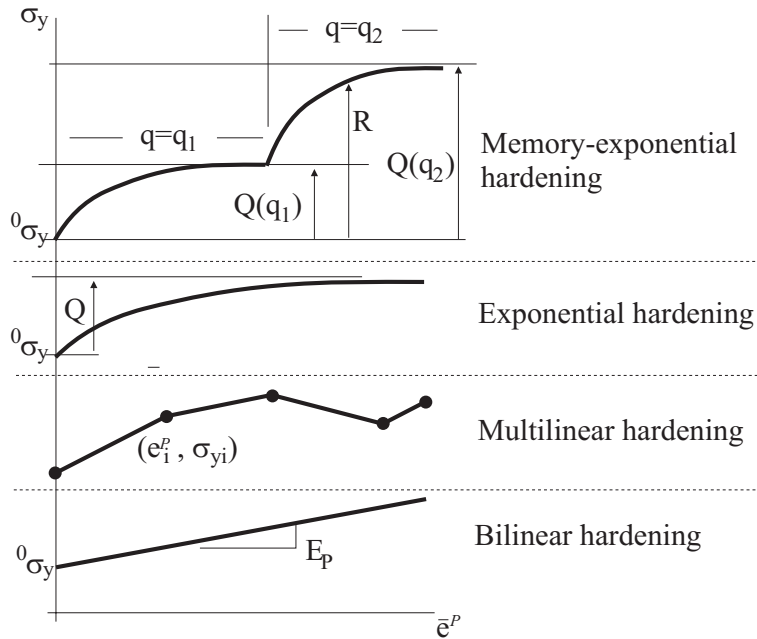


Figure 3.4-18: Dependence of stress radius on accumulated effective plastic strain

Bilinear hardening:

$${}^{t+\Delta t}\sigma_y = {}^0\sigma_y + E_p {}^{t+\Delta t}\bar{\epsilon}^p$$

where ${}^0\sigma_y$ and E_p are material constants, and ${}^{t+\Delta t}\bar{\epsilon}^p$ is the accumulated effective plastic strain (p in ref LC). ${}^{t+\Delta t}\bar{\epsilon}^p$ is calculated using ${}^{t+\Delta t}\bar{\epsilon}^p = {}^t\bar{\epsilon}^p + \Delta\bar{\epsilon}^p$, where $\Delta\bar{\epsilon}^p = \sqrt{\frac{2}{3}}\Delta\mathbf{e}^p : \Delta\mathbf{e}^p$.

For bilinear hardening, $E_p = \frac{EE_T}{E - E_T}$ where E_T is the slope of the stress-strain curve during plasticity (this formula assumes no kinematic hardening).

Multilinear hardening:

Pairs of (\bar{e}^P, σ_y) are given. It is allowed for σ_y to either increase or decrease as \bar{e}^P increases.

If the multilinear hardening curve is given in terms of the uniaxial response (e, σ) where σ is the tensile stress corresponding to the uniaxial strain e , then

$\left(\bar{e}^P = e - \frac{\sigma}{E}, \sigma_y = \sigma \right)$ are the corresponding points for the points (e, σ) . This formula assumes no kinematic hardening.

Exponential hardening:

$${}^{t+\Delta t}\sigma_y = {}^0\sigma_y + Q(1 - \exp(-b {}^{t+\Delta t}\bar{e}^P))$$

where Q and b are material constants. Q can be positive to model cyclic hardening, and Q can be negative to model cyclic softening.

Memory-exponential hardening (exponential hardening with strain memory surface):

In this model, the yield surface size ${}^{t+\Delta t}\sigma_y$ depends on the strain memory size ${}^{t+\Delta t}q$ (see above for a description of the strain memory surface).

The yield surface size is

$${}^{t+\Delta t}\sigma_y = {}^0\sigma_y + {}^{t+\Delta t}R$$

where ${}^{t+\Delta t}R$ is the change in the size of the yield surface. ${}^{t+\Delta t}R$ is defined using the differential equation

$$dR = b({}^tQ - {}^tR)d\bar{e}^P$$

and is calculated using ${}^{t+\Delta t}R = {}^tR + \Delta R$,
 $\Delta R = b({}^{t+\Delta t}Q - {}^{t+\Delta t}R)\Delta\bar{e}^P$. ${}^{t+\Delta t}Q$ is the asymptotic change in
yield surface size, and is calculated as

$${}^{t+\Delta t}Q = Q({}^{t+\Delta t}q) = Q_M - (Q_M - Q_0)\exp(-2\mu {}^{t+\Delta t}q)$$

where ${}^{t+\Delta t}q$ is the size of the strain memory surface.

The material constants for the memory-exponential material model
are ${}^0\sigma_y$, Q_0 , Q_M , b , μ and the strain memory surface
parameter η .

Kinematic hardening rule: The kinematic hardening rule is:

Armstrong-Fredrick nonlinear kinematic hardening:

The back stress is expressed as a sum of partial back stresses

$${}^{t+\Delta t}\boldsymbol{\alpha} = \sum {}^{t+\Delta t}\boldsymbol{\alpha}^{(m)}$$

where ${}^{t+\Delta t}\boldsymbol{\alpha}^{(m)}$ is partial back stress number m . All of the partial
back stresses are independent of each other.

Each partial back stress evolves according to the differential rule

$$d\boldsymbol{\alpha}^{(m)} = \frac{2}{3}h^{(m)}d\mathbf{e}^P - \zeta^{(m)} {}^t\boldsymbol{\alpha}^{(m)} d\bar{e}^P$$

where $h^{(m)}$ and $\zeta^{(m)}$ are material constants (these material
constants are C_l and γ_l for partial back stress number l in ref LC).

Assuming that the directions of plastic strain increments are
constant during a time step, this can be integrated to obtain

$${}^{t+\Delta t}\boldsymbol{\alpha}^{(m)} = \mathbf{A}^{(m)} - (\mathbf{A}^{(m)} - {}^t\boldsymbol{\alpha}^{(m)})\exp(-\Delta b^{(m)})$$

where $\mathbf{A}^{(m)} = \sqrt{\frac{2}{3}} \frac{h^{(m)}}{\zeta^{(m)}} (\text{directions of } \Delta \mathbf{e}^P)$, $\Delta b^{(m)} = \zeta^{(m)} \Delta \bar{e}^P$.

It is allowed to use one partial back stress, with $\zeta = 0$. Then linear kinematic hardening is recovered, with $E_p = h$ (this formula assumes no isotropic hardening).

Stress – plastic strain curve for uniaxial cycling

In uniaxial cycling, a typical stress – plastic strain curve is shown in Figure 3.4-19. Here only one partial back stress is used with material constants h and ζ . Since a stable cycle is considered here, σ_y is taken from the isotropic hardening rule assuming a very large value of \bar{e}^P . The hardening modulus is given by $E_p = h - \zeta (\sigma - \sigma_y)$ on segment A-B. The size of the plastic strain cycle can be related to the material constants using the relation

$$\frac{\Delta \sigma}{2} = \frac{h}{\zeta} \tanh \left(\zeta \frac{\Delta e^P}{2} \right) + \sigma_y$$

Thus, given several cycles for different cyclic strains, the material constants can be estimated using the above formula.

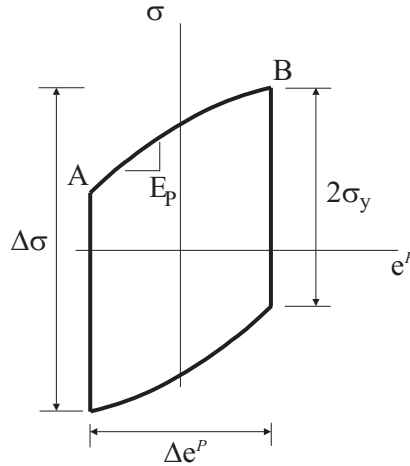


Figure 3.4-19: Stable uniaxial plastic cycle using one term in Armstrong-Fredrick nonlinear kinematic hardening rule

Stress – plastic strain curve for initial loading

During initial loading, it can be shown that

$$\sigma = \sigma_y + \frac{h}{\zeta} \left(1 - \exp(-\zeta e^p) \right)$$

The corresponding stress – plastic strain curve is shown in Fig. 3.4-20. Here only one partial back stress is used with material constants h and ζ . In this formula, it is assumed that there is no isotropic hardening, and that the yield stress is σ_y . Hence the hardening observed during initial loading can be modeled using nonlinear kinematic hardening, without the use of isotropic hardening.

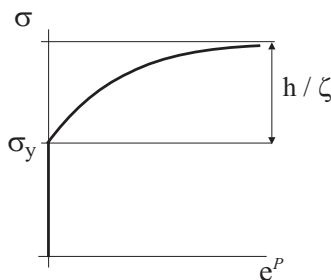


Figure 3.4-20: Stress - plastic strain curve during initial loading, using nonlinear kinematic hardening

Combination of isotropic and kinematic hardening rules: By combining the Armstrong-Fredrick nonlinear kinematic hardening rule with the isotropic hardening rules, a wide variety of cyclic phenomena can be simulated, such as cyclic hardening and softening, shakedown and ratcheting. See ref LC for details.

Stress-strain integration: When plasticity is detected, the incremental plastic strains are solved for using an iterative solution procedure. The maximum number of iterations can be specified, and the tolerance used in assessing convergence can be specified.

Constitutive tensor: The constitutive tensor (stress-strain matrix) is constructed to be tangent. This gives good convergence in the equilibrium iterations. However, in general, the tangent constitutive tensor is non-symmetric. Therefore it is allowed to use the non-symmetric equation solver when the plastic-cyclic model is used. It is also allowed to use a symmetric equation solver; in this case, the constitutive tensor is symmetrized, and this will reduce the rate of convergence (increase the number of equilibrium iterations).

Formulations: When used with the small displacement/small strain formulation, a materially-nonlinear-only formulation is employed, when used with the large displacement/small strain formulation, a TL is employed, and when used with the large displacement/large strain formulation, either a ULH or ULJ formulation is employed.

Mixed displacement-pressure formulation: Plane strain, axisymmetric and 3-D elements that reference these material models should also employ the mixed displacement-pressure (u/p) element formulation. This is because the plastic strains are incompressible. The u/p formulation is the default for these elements.

Thermal strains: If a thermal load is applied to the structure, the thermal strains are taken into account but the material characteristics are considered to be temperature independent.

Modeling of rupture: Rupture conditions can also be included with this model.

The maximum accumulated effective plastic strain can be specified for the rupture condition. When rupture is reached at a given element integration point, the corresponding element is removed from the model (see Section 11.5).

The option of user-supplied rupture is not available for the plastic-cyclic material.

Rate-dependent plasticity: The option of rate-dependent plasticity is not available for the plastic-cyclic model.

3.4.5.2 Specification of input

MATERIAL PLASTIC-CYCLIC: The basic command is

```
MATERIAL PLASTIC-CYCLIC NAME E NU DENSITY,  
      ALPHA PLCYCL-ISOTROPIC PLCYCL-KINEMATIC,  
      PLCYCL-RUPTURE BETA MAXITE RTOL
```

This command references data defined in commands `PLCYCL-ISOTROPIC`, `PLCYCL-KINEMATIC`, `PLCYCL-RUPTURE`. These commands provide the input for the isotropic, kinematic and rupture parts of the plastic-cyclic model. Each of the isotropic, kinematic and rupture options given above has a corresponding `PLCYCL-ISOTROPIC`, `PLCYCL-KINEMATIC` and `PLCYCL-RUPTURE` command, see below for a detailed description of these commands.

It is allowed to not specify a value for *PLCYCL-KINEMATIC* or *PLCYCL-RUPTURE*. Then these effects are not included in the model.

BETA is the integration factor β described above. When BETA=AUTOMATIC, ADINA chooses the value of β as follows: $\beta=1$ for static or implicit time integration; $\beta=0$ for explicit time integration. MAXITE is the maximum number of iterations used to solve for the incremental plastic strains. RTOL is a tolerance used to assess convergence of the iterations. RTOL can be thought of as a reference incremental plastic strain.

PLCYCL-ISOTROPIC: The commands

```
PLCYCL-ISOTROPIC BILINEAR NAME YIELD EP
```

```
PLCYCL-ISOTROPIC MULTILINEAR NAME  
aepsi stress-radiusi
```

```
PLCYCL-ISOTROPIC EXPONENTIAL NAME YIELD Q B
```

```
PLCYCL-ISOTROPIC MEMORY-EXPONENTIAL NAME,  
YIELD Q0 QM MU B ETA
```

are used to specify the constants for the isotropic part of the plastic-cyclic model.

PLCYCL-KINEMATIC: The command

```
PLCYCL-KINEMATIC ARMSTRONG-FREDRICK NAME  
hi zetai
```

is used to specify the constants for the kinematic part of the plastic-cyclic model.

PLCYCL-RUPTURE: The command

```
PLCYCL-RUPTURE AEPS NAME VALUE
```

is used to specify the constants for the rupture part of the plastic-

cyclic model. Rupture is based on the accumulated effective plastic strain.

Input examples:

1) The simplest material that can be defined using this command is given by a command sequence such as

```
PLCYCL-ISOTROPIC BILINEAR 1 YIELD=2E8  
MATERIAL PLASTIC-CYCLIC 1 E=2.07E11 NU=0.3,  
DENSITY=7800 PLCYCL-ISOTROPIC=1
```

This material is perfectly plastic. This material description is equivalent to

```
MATERIAL PLASTIC-BILINEAR 1 E=2.07E11 NU=0.3,  
DENSITY=7800 YIELD=2E8
```

2) The PLASTIC-CYCLIC material can be used to model bilinear isotropic hardening using a command sequence such as

```
PLCYCL-ISOTROPIC BILINEAR 1 YIELD=2E8,  
EP=2.090909E09  
MATERIAL PLASTIC-CYCLIC 1 E=2.07E11 NU=0.3,  
DENSITY=7800 PLCYCL-ISOTROPIC=1
```

and this material description is equivalent to

```
MATERIAL PLASTIC-BILINEAR 1 E=2.07E11 NU=0.3,  
DENSITY=7800 YIELD=2E8 ET=2.07E09
```

3) The plastic-cyclic material can be used to model multilinear isotropic hardening using a command sequence such as

```
PLCYCL-ISOTROPIC MULTILINEAR 1  
0 2.0E8  
1E-3 2.5E8  
2E-3 2.7E8  
MATERIAL PLASTIC-CYCLIC 1 E=2.07E11 NU=0.3,  
DENSITY=7800 PLCYCL-ISOTROPIC=1
```

and this material description is equivalent to

```
MATERIAL PLASTIC-MULTILINEAR 1 E=2.07E11,  
    NU=0.3  
9.6618E-4 2.0E8  
2.2077E-3 2.5E8  
3.3043E-3 2.7E8
```

4) The plastic-cyclic material can be used to model bilinear kinematic hardening using a command sequence such as

```
PLCYCL-ISOTROPIC BILINEAR 1 YIELD=2E8  
PLCYCL-KINEMATIC ARMSTRONG-FREDRICK 1  
2.090909E9  
MATERIAL PLASTIC-CYCLIC 1 E=2.07E11 NU=0.3,  
    DENSITY=7800 PLCYCL-ISOTROPIC=1,  
    PLCYCL-KINEMATIC=1
```

and this material description is equivalent to

```
MATERIAL PLASTIC-BILINEAR 1 E=2.07E11 NU=0.3,  
    DENSITY=7800 YIELD=2E8 ET=2.07E09,  
    HARDENING=KINEMATIC
```

Conversion formulas

Plastic-bilinear to plastic-cyclic:

In general, given a plastic-bilinear material of the form

```
MATERIAL PLASTIC-BILINEAR ...,  
    HARDENING=ISOTROPIC,  
    E=E YIELD=YIELD ET=ET
```

this material can be converted into an equivalent plastic-cyclic material as follows:

```
PLCYCL-ISOTROPIC BILINEAR 1 YIELD=YIELD EP=EP  
MATERIAL PLASTIC-CYCLIC... E=E... ,  
    PLCYCL-ISOTROPIC=1
```

using the formula $EP = \frac{E E_T}{E - E_T}$.

Plastic-multilinear to plastic-cyclic:

In general, given a plastic-multilinear material of the form

```
MATERIAL PLASTIC-MULTILINEAR ...,
                                HARDENING=ISOTROPIC,
                                E=E

strain1  stress1
strain2  stress2
...
straini  stressi
```

this material can be converted into an equivalent plastic-cyclic material as follows:

```
PLCYCL-ISOTROPIC MULTILINEAR 1
aeps1  stress-radius1
aeps2  stress-radius2
...
aepsi  stress-radiusi
MATERIAL PLASTIC-CYCLIC ... E=E ...,
        PLCYCL-ISOTROPIC=1
```

using the formulas

$$\begin{aligned} aeps_i &= strain_i - (stress_i / E) \\ stress-radius_i &= stress_i \end{aligned}$$

Note that $aeps_1$ will always be zero using these formulas.

3.4.5.3 Output variables

The material model outputs stresses, strains, plastic strains, accumulated effective plastic strain ${}^{t+\Delta t}\bar{\epsilon}^P$, yield stress ${}^{t+\Delta t}\sigma_y$,

stress radius (defined as $\sqrt{\frac{3}{2}} \| {}^{t+\Delta t}\mathbf{s} - {}^{t+\Delta t}\boldsymbol{\alpha} \|$) and dissipated plastic

work (defined as $w_p = \int_0^{t+\Delta t} \boldsymbol{\tau} : d\mathbf{e}^P$).

When there are thermal effects, thermal strains are also output.
When the strain memory surface is used, the strain memory size ${}^{t+\Delta t}q$ is also output.

3.5 Thermo-elastic material models

- The thermo-isotropic and thermo-orthotropic material models are used to calculate stress distributions due to imposed temperatures. For thermal stress analysis, the material moduli can be constant or temperature-dependent.
- The thermo-isotropic model is available for the **truss**, **2-D solid**, **3-D solid**, **iso-beam**, **shell** and **pipe** elements.
- The thermo-orthotropic model is available for the **2-D solid**, **3-D solid** and **shell** elements.
- Both models can be used with the **small displacement** and **large displacement** formulations. In all cases the strains are assumed to be small.

When used with the small displacement formulation, a materially-nonlinear-only formulation is employed, and when used with the large displacement formulation, either the TL or UL formulation is employed.

- In the data input for the analysis, the nodal point temperatures must be defined for all time steps (calculated by the ADINA heat flow solution option or reading these temperatures from a file created by ADINA-T, reading these temperatures from a mapping file or defining them by time functions). If the model includes axisymmetric shell or shell elements, the nodal point temperature gradients must also be defined for all time steps. See Section 5.6 for more information on prescribing temperatures and temperature gradients.
- For these models, the elastic moduli, the shear moduli, the Poisson's ratios and the coefficients of thermal expansion are input as piecewise linear functions of the temperature, as illustrated in Fig. 3.5-1. Linear interpolation is used to calculate the material properties between input points.

- For calculation of thermal strains, please refer to Section 3.1.4.

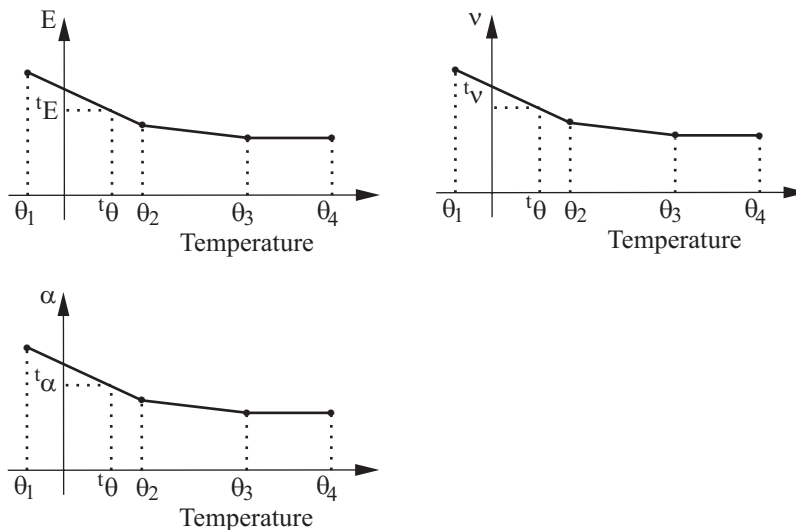


Figure 3.5-1: Variation of material properties for thermo-elastic model

- For the evaluation of the temperatures ${}^t\theta$ and ${}^0\theta$ at the integration point considered, the isoparametric interpolation functions h_i are used; e.g., in two-dimensional analysis we have

$$\theta = \sum_{i=1}^q h_i \theta_i$$

where θ_i is the temperature at element nodal point i (see Fig. 3.5-2). Note that when higher-order elements are used the temperatures at the integration point can be significantly different from the values at the nodal point (for example negative although all nodal point temperatures are greater than or equal to zero).

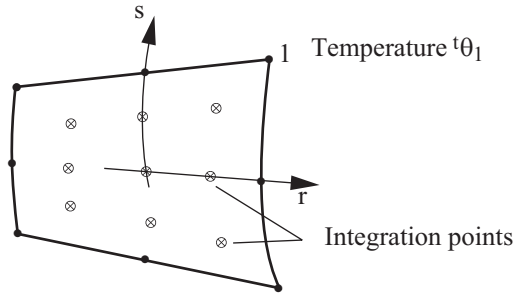


Figure 3.5-2: Interpolation of temperature at integration points

- For axisymmetric shell and shell elements, the temperature at an integration point is established using the temperatures and temperature gradients at the midsurface nodes. Note that the temperature gradient at a shell midsurface node is defined in the direction of the mid-surface director vector at that node, see Section 2.7.3.

For transition shell elements, no temperature gradients are associated with the top/bottom nodes and only the temperatures at the top/bottom nodes are used to evaluate the temperatures at the integration points.

- In the thermo-orthotropic material model, the coefficients of thermal expansion can be different in the three material directions.

3.6 Thermo-elasto-plasticity and creep material models

3.6.1 Thermo-elastic-plastic-creep material models

- This section describes the **thermo-plastic, creep, plastic-creep, multilinear-plastic-creep, creep-variable, plastic-creep-variable, multilinear-plastic-creep-variable** and **irradiation creep** material models.

ref. KJB
Section 6.6.3

- The thermo-elasto-plasticity and creep models include the effects of

- ▶ Thermal strains, ${}^t e_{rs}^{TH}$

- ▶ Time-independent plastic strains, ${}^t e_{rs}^P$
- ▶ Time-dependent creep strains, ${}^t e_{rs}^C$

and the constitutive relation used is

$${}^t \sigma_{ij} = {}^t C_{ijrs}^E \left({}^t e_{rs} - {}^t e_{rs}^P - {}^t e_{rs}^C - {}^t e_{rs}^{TH} \right)$$

where ${}^t \sigma_{ij}$ is the stress tensor at time t and ${}^t C_{ijrs}^E$ is the elasticity tensor at the temperature corresponding to time t . The tensor ${}^t C_{ijrs}^E$ can be expressed in terms of Young's modulus ${}^t E$ and Poisson's ratio ${}^t \nu$ which are both temperature-dependent.

- The irradiation creep model includes the effects of

- ▶ Thermal strains, ${}^\phi e_{rs}^{TH}$
- ▶ Irradiation strains, ${}^\phi e_{rs}^W$
- ▶ Neutron fluence-dependent creep strains, ${}^\phi e_{rs}^C$

and the constitutive relation used is

$${}^\phi \sigma_{ij} = {}^\phi C_{ijrs}^E \left({}^\phi e_{rs} - {}^\phi e_{rs}^C - {}^\phi e_{rs}^W - {}^\phi e_{rs}^{TH} \right)$$

where ${}^\phi \sigma_{ij}$ is the stress tensor corresponding to the neutron fluence ${}^\phi$ at time t and ${}^\phi C_{ijrs}^E$ is the elasticity tensor corresponding to the neutron fluence ${}^\phi$ at time t . The tensor ${}^\phi C_{ijrs}^E$ can be expressed in terms of Young's modulus ${}^\phi E$ which also corresponds to the neutron fluence ${}^\phi$ at time t .

- These material models can be used with the **truss**, **2-D solid**, **3-D solid**, **iso-beam**, **shell** and **pipe** elements, with the exception of the creep-irradiation model which can only be used with **2-D solid** and **3-D solid** elements.

- These models can be used with the **small displacement/small strain, large displacement/small strain and large displacement/large strain** formulations. The large displacement/large strain formulation can only be used with the 2-D solid and 3-D solid elements.

When used with the small displacement/small strain formulation, a materially-nonlinear-only formulation is employed, when used with the large displacement/small strain formulation, either a TL or a UL formulation is employed and when used with the large displacement/large strain formulation, the ULH formulation is employed.

- If geometrically nonlinear effects are to be included, the large displacements/large strain kinematics are preferred to the large displacement/small strain kinematics, even when the strains are numerically small. The large displacement/small strain kinematics should be used only when the large displacement/large strain kinematics are not supported by the element. By default large strain kinematics are used for 2-D solid and 3-D solid elements when large displacement kinematics are selected.
- Plane strain, axisymmetric or 3-D solid elements that reference these material models should also employ the mixed displacement-pressure (u/p) element formulation. The u/p formulation is the default for these elements.
- Note that the constitutive relations for the thermal, plastic and creep strains are independent of each other; hence the only interaction between the strains comes from the fact that all strains affect the stresses. Figure 3.6-1 summarizes the constitutive description for a one-dimensional stress situation and a bilinear stress-strain curve.

Since there is no direct coupling in the evaluation of the different strain components, we can discuss the calculation of each strain component independently.

ref. M.D. Snyder and K.J. Bathe, "A Solution Procedure for Thermo-Elastic-Plastic and Creep Problems," *J. Nuclear*

Eng. and Design, Vol. 64, pp. 49-80, 1981.

ref. M. Kojić and K.J. Bathe, "The Effective-Stress-Function Algorithm for Thermo-Elasto-Plasticity and Creep," *Int. J. Numer. Meth. Engng.*, Vol. 24, No. 8, pp. 1509-1532, 1987.

- Rate-dependent plasticity described in Section 3.4 can also be applied to thermoplasticity models with isotropic hardening (bilinear or multilinear).
- Rupture of the material based on the following three-invariant criterion can be specified:

$$\alpha J_3({}'\sigma) + \beta J_1({}'\sigma) + (1 - \alpha - \beta) J_2({}'\sigma) \leq \bar{\sigma}_u$$

in which ${}'\sigma$ is the stress tensor, J_1 , J_2 , J_3 are the first, second and third stress tensor invariants, α , β are material properties (input data) and $\bar{\sigma}_u$ is the ultimate effective stress (input data).

This rupture criteria can be used in combination with other rupture criteria described later in this section.

When rupture is reached at a given element integration point, the corresponding element is removed from the model (see Section 11.5).

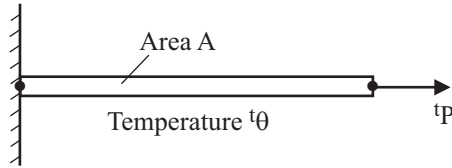
3.6.1.1 Evaluation of thermal strains

- The thermal strains are calculated from the prescribed nodal point temperatures with the coefficient of thermal expansion being temperature dependent. The procedure used is the same as for the thermo-isotropic material model, see Section 3.5.
- The thermal strains of the irradiation creep model are dependent on temperature and neutron fluence. The value of thermal strain is determined using the Shifting rule described in Section 3.6.5.
- For the multilinear plastic-creep material model, there is the

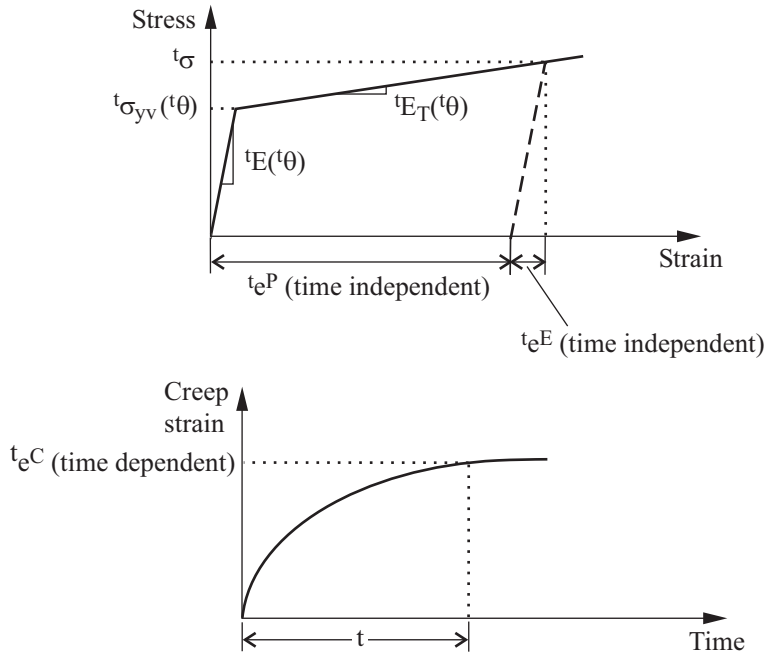
option of entering orthotropic thermal coefficients, in which the coefficients of thermal expansion are different in the three material direction. However, this option cannot be used in conjunction with the large strain formulations in 2D and 3D elements.

3.6.1.2 Evaluation of plastic strains

- Plasticity effects are included in the thermo-plastic, plastic-creep, multilinear-plastic-creep, plastic-creep-variable and multilinear-plastic-creep-variable material models. For the material models that have the word "multilinear" in their names, the stress-strain curve is multilinear, otherwise the stress-strain curve is bilinear.



(a) Model problem of truss element under constant load



(b) Strains considered in the model

Figure 3.6-1: Thermo-elasto-plasticity and creep constitutive description in one-dimensional analysis

- The plastic strains are calculated using the von Mises plasticity model (see Section 3.4.1) with temperature-dependent material parameters (Young's modulus, strain hardening modulus, Poisson's ratio, yield stress, etc...). Figure 3.6-2 shows the assumptions used for the dependence of the material parameters on temperature (in the case of a bilinear stress-strain law). (Note that the plastic

modulus

$$E_p = \frac{E E_T}{E - E_T}$$

is interpolated as a function of temperature, not the tangent modulus E_T .) Hence, the yield function is, for isotropic hardening

$${}^t f_y = \frac{1}{2} {}^t \mathbf{s} \cdot {}^t \mathbf{s} - \frac{1}{3} {}^t \sigma_{yv}^2$$

and for kinematic hardening

$${}^t f_y = \frac{1}{2} ({}^t \mathbf{s} - {}^t \mathbf{a}) \cdot ({}^t \mathbf{s} - {}^t \mathbf{a}) - \frac{1}{3} {}^t \sigma_{yv}^2$$

where ${}^t \mathbf{s}$ is the deviatoric stress tensor, ${}^t \sigma_{yv}$ is the virgin yield stress corresponding to temperature ${}^t \theta$ (see Fig. 3.6-2) and ${}^t \mathbf{a}$ is the shift of the stress tensor due to kinematic hardening. In the case of multilinear stress-strain curves, the yield curves are interpolated during the solution as shown in Fig. 3.6-3.

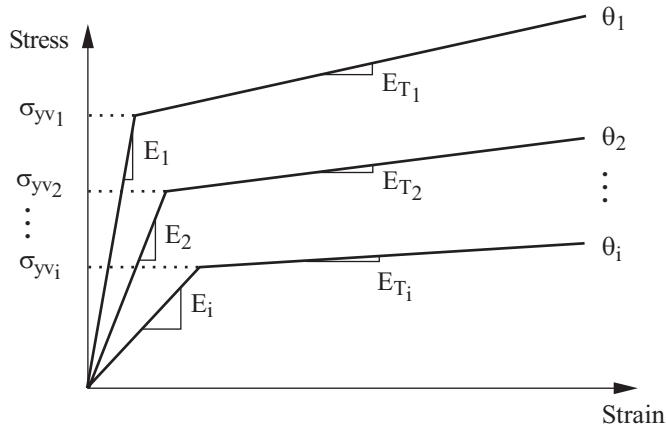
- The expressions for plastic strain increments resulting from the flow theory are $de_{ij}^p = d\lambda {}^t s_{ij}$ for isotropic hardening and $de_{ij}^p = d\lambda ({}^t s_{ij} - {}^t \alpha_{ij})$ for kinematic hardening, in which $d\lambda$ is the plastic multiplier (positive scalar) which can be determined from the yield condition ${}^t f_y = 0$. In the case of kinematic hardening, we express the change of the yield surface position in the form

$$d\alpha_{ij} = {}^t C de_{ij}^p$$

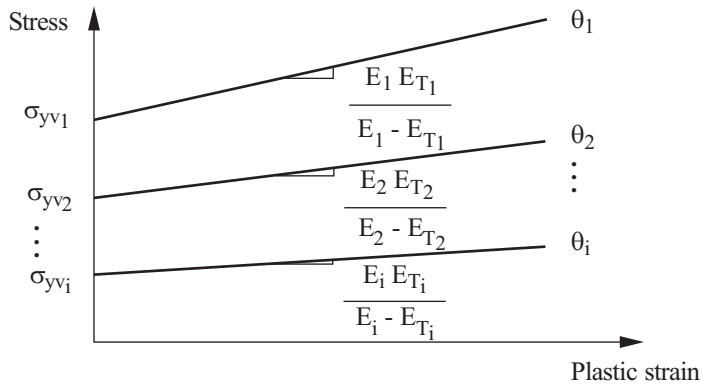
where ${}^t C$ is the modulus

$${}^t C = \frac{2}{3} {}^t E_p = \frac{2}{3} \frac{{}^t E {}^t E_T}{{}^t E - {}^t E_T}$$

In the case of multilinear yield curves, ${}^t E_T$ represents the tangent modulus of the segment on the yield curve corresponding to the accumulated effective plastic strain ${}^t \bar{\epsilon}^p$.

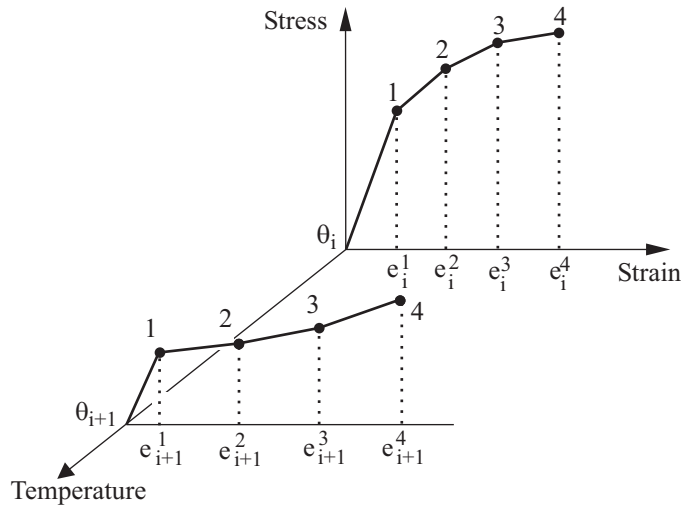


(a) Uniaxial relation of stress versus total strain

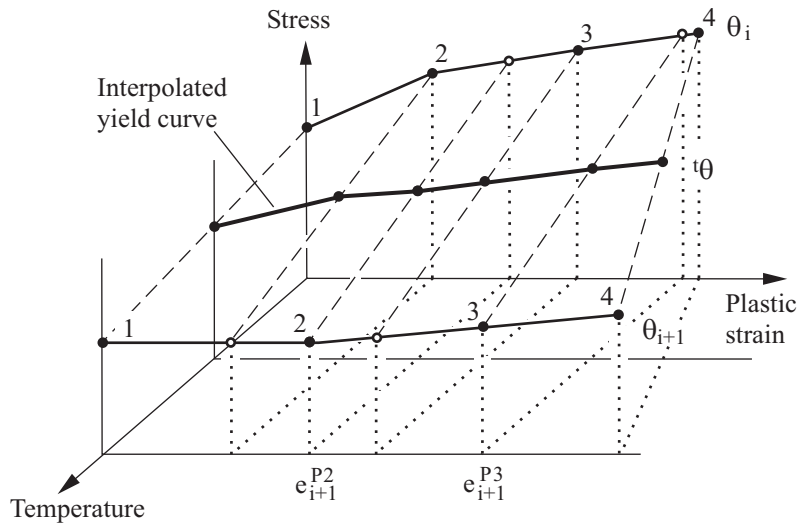


(b) Uniaxial relation of stress versus plastic strain

Figure 3.6-2: Variation of material properties with temperature for the thermo-elasto-plasticity and creep models; bilinear stress-strain curves



a) Stress-strain curves input data



b) Yield curves

Figure 3.6-3: Interpolation of multilinear yield curves with temperature

- The formula $d\alpha_{ij} = {}^tC de_{ij}^P$ requires modification to avoid nonphysical effects when tE_p , and hence tC , is a function of temperature.

Here is an example showing a nonphysical effect that occurs when tE_p is a function of temperature in kinematic hardening. Consider the following material description, in which kinematic hardening is used:

Temperature	E	ν	σ_{yv}	E_T	E_p
0	1E6	0	100	1996.008	2000
200	1E6	0	100	999.001	1000

The stress-plastic strain curves for this material are shown in Figure 3.6-4. Take a uniaxial specimen, set the temperature to 0 and load with prescribed force until the uniaxial stress is 130. The corresponding plastic strain is $e^P = 0.015$ (since $100 + 2000e^P = 130$). This point is labeled A in Figure 3.6.4.

Now, without changing the prescribed force, change the temperature to 200. For the plastic strain $e^P = 0.015$, the yield stress is $100 + 1000e^P = 115$, which is less than the stress of 130. So we would expect that the material would plastically deform further until the (plastic strain, stress) point is on the yield curve corresponding to temperature 200 (point B in Figure 3.6-4). However the plastic strain does not change and the current stress remains above the yield curve for temperature 200 (point A).

The reason for this unexpected behavior is as follows: The yield condition involves the initial yield stress, which is unchanged, the current stress, which is unchanged and the back stress, which is also unchanged (if the back stress had changed, then the yield condition would not be satisfied). Since the change in plastic strain is related to the change in back stress through $d\alpha_{ij} = {}^tC de_{ij}^P$, because the back stress does not change, the plastic strain does not change either.

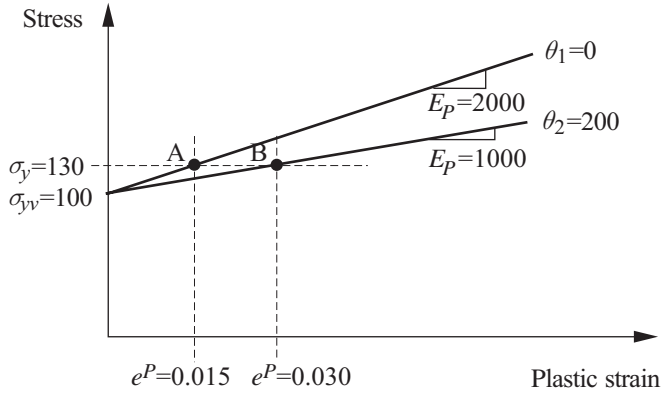


Figure 3.6-4: Example showing nonphysical results in kinematic hardening when E_p is a function of temperature

- Chaboche has discussed the formulation of kinematic hardening under varying temperature conditions, and has pointed out other nonphysical effects when the above back stress evolution rule is used in varying temperature conditions, see Section 4.4 of the following reference:

ref. J.L. Chaboche, “Cyclic viscoplastic constitutive equations, part I: A thermodynamically consistent formulation”, *J. Appl. Mech*, December 1993, Vol 60, pp 813-821.

For bilinear kinematic hardening, in which tC is a function of temperature, but in which tC is not a function of plastic strain, Chaboche suggests a rule equivalent to

$${}^t\alpha_{ij} = {}^tC {}^te_{ij}^P$$

which in differential form becomes

$$d\alpha_{ij} = {}^tC de_{ij}^P + {}^te_{ij}^P dC$$

It is seen that the first term on the right-hand-side corresponds to the formulation presented above. The second term on the right-hand side corresponds to a “back stress temperature correction”, since this term is nonzero only if the temperature is varying.

For multilinear kinematic hardening, in which $'C$ is a function of temperature and of plastic strain, we again include a temperature correction corresponding to the second term on the right-hand side, but in which dC in the second term is evaluated only due to changes in temperature. With this correction, multilinear kinematic hardening with constant stress - plastic strain slope gives the same results as bilinear kinematic hardening.

When the temperature correction is included in the above example, the plastic strain in the above example increases to point B when the temperature is increased to 200, so that the current stress remains on the yield curve. This response is the expected behavior.

The back stress temperature correction can be selected using the AUI command

```
KINEMATICS,  
  BACKSTRESS-TEMPERATURE-CORRECTION=YES
```

The default is to use the back stress temperature correction.

- Rupture due to plasticity can be modeled using this material model. When using a bilinear stress-strain curve, a maximum allowable effective plastic strain \bar{e}_A^P can be input. When using a multilinear stress-strain curve, the maximum allowable effective plastic strain at a given temperature corresponds to the last point on the yield curve at that temperature (see Fig. 3.6-3).

3.6.1.3 Evaluation of creep strains

Creep, plastic-creep, multilinear-plastic-creep material models

- The effective creep strain is calculated using one of the following creep laws:

Power creep law (creep law 1) :

$${}^t\bar{\epsilon}^C = a_0 {}^t\sigma^{a_1} t^{a_2}$$

in which a_0, a_1, a_2 are material constants.

Exponential creep law (creep law 2) :

$${}^t\bar{\epsilon}^C = F(1 - e^{-Rt}) + Gt$$

with

$$F = a_0 e^{a_1 \cdot {}^t\sigma}; \quad R = a_2 \left(\frac{{}^t\sigma}{a_3} \right)^{a_4}; \quad G = a_5 e^{a_6 \cdot {}^t\sigma}$$

in which a_0 to a_6 are material constants.

Eight-parameter creep law (creep law 3) :

$${}^t\bar{\epsilon}^C = S \cdot T \cdot e^{-H}$$

with

$$S = a_0 {}^t\sigma^{a_1}; \quad T = t^{a_2} + a_3 t^{a_4} + a_5 t^{a_6}, \quad H = \frac{a_7}{{}^t\theta}$$

in which a_0 to a_7 are material constants.

In the above equations, ${}^t\bar{\epsilon}^C$, ${}^t\sigma$ and ${}^t\theta$ denote the effective creep strain, effective stress and temperature at time t . The unit of the temperature can either be K or °C (default). ADINA automatically applies the conversion to K in the default setting.

LUBBY2 creep law: The LUBBY2 creep law can be employed to predict long-term behavior of rock salt mass.

The LUBBY2 creep law is implemented for 2-D solid and 3-D solid elements. It can be used in conjunction with strain-hardening or time-hardening.

The LUBBY2 creep law is specified by the following equations:

$$\frac{d^t\bar{\epsilon}^C}{dt} = \left[\frac{1}{\bar{\eta}_K(^t\bar{\sigma})} \exp\left(-\frac{\bar{G}_K(^t\bar{\sigma})}{\bar{\eta}_K(^t\bar{\sigma})}t\right) + \frac{1}{\bar{\eta}_M(^t\bar{\sigma})} \right] ^t\bar{\sigma}$$

or

$$^t\bar{\epsilon}^C = \left[\frac{-1}{\bar{G}_K(^t\bar{\sigma})} \exp\left(-\frac{\bar{G}_K(^t\bar{\sigma})}{\bar{\eta}_K(^t\bar{\sigma})}t\right) + \frac{t}{\bar{\eta}_M(^t\bar{\sigma})} + \frac{1}{\bar{G}_K(^t\bar{\sigma})} \right] ^t\bar{\sigma}$$

where $\bar{\eta}_M(^t\bar{\sigma}) = \bar{\eta}_M^* e^{m^t\bar{\sigma}}$, $\bar{G}_K(^t\bar{\sigma}) = \bar{G}_K^* e^{k_1^t\bar{\sigma}}$ and

$\bar{\eta}_K(^t\bar{\sigma}) = \bar{\eta}_K^* e^{k_2^t\bar{\sigma}}$. Also, as usual, $^t\bar{\epsilon}^C$ is the effective creep

strain, $\frac{d^t\bar{\epsilon}^C}{dt}$ is the effective creep strain rate and $^t\bar{\sigma}$ is the effective creep stress.

In these equations, $m, k_1, k_2, \bar{\eta}_M^*, \bar{G}_K^*$ and $\bar{\eta}_K^*$ are six material parameters. m, k_1, k_2 account for the stress effect on parameters $\bar{\eta}_M^*, \bar{G}_K^*, \bar{\eta}_K^*$. The six material parameters are identified in ADINA as a0, a1, a2, a3, a4 and a5 respectively.

Unloading and cyclic loading effects are considered in the LUBBY2 creep law.

Analytical solutions with the LUBBY2 creep law are very sensitive to the three parameters m, k_1, k_2 , so these parameters must be very carefully specified.

For more information about the LUBBY2 creep law, see the following reference:

ref. K.H. Lux and S. Heusermann, “Creep Tests on Rock Salt with Changing Load as a Basis for the Verification of Theoretical Material Laws”, *Proc. 6th Symp. on Salt*, Toronto, Vol. 1, pp. 417-435, 1983.

Blackburn creep law:

$${}^t\bar{\epsilon}^C = C_1 \{1 - \exp(-r_1 t)\} + C_2 \{1 - \exp(-r_2 t)\} + Gt$$

with

$$C_1 = \frac{A_0 G^{A_1}}{r_1}, \quad C_2 = \frac{A_2 G^{A_3}}{r_2}$$

$$r_1 = A_4 t_R^{-A_5}, \quad r_2 = A_6 t_R^{-A_7}, \quad G = F t_R^{-A_8}$$

$$\log_{10}(t_R) = B_0 + B_1 \log_{10} {}^t\sigma + B_2 (\log_{10} {}^t\sigma)^2 + A_9$$

$$F = A_{10} \exp\left(\frac{A_{11}}{{}^t\theta + 273.16}\right)$$

$$B_0 = \frac{A_{12}}{{}^t\theta + 273.16} + A_{13}, \quad B_1 = \frac{A_{14}}{{}^t\theta + 273.16}, \quad B_2 = \frac{A_{15}}{{}^t\theta + 273.16}$$

$A_0, A_1, A_2, \dots, A_{15}$ are material constants. The unit of temperature should be Celsius.

Creep-variable, plastic-creep-variable, multilinear-plastic-creep-variable material models

The following creep laws can be employed:

Eight-parameter creep law (creep law 3) with variable coefficients: The same functional form as in creep law 3 described above is employed, but coefficients a_0 to a_7 are considered to be functions of temperature and effective stress.

The values of a_0 to a_7 can also be evaluated within user-supplied subroutines. In the user-supplied subroutines, these parameters can be functions of the temperature and the effective stress.

The user-supplied subroutines are:

UCOEF2 for 2-D solid elements (file ovl30u.f)
 UCOEF3 for 3-D solid elements (file ovl40u.f)
 UCOEFB for iso-beam elements (file ovl60u.f)
 UCOEFS for shell elements (file ovl100u.f)
 UCOEFP for pipe elements (file ovl110u.f)

These subroutines are called by ADINA every time the stress-strain law is updated.

Six-parameter LUBBY2 creep law with variable coefficients:

The same functional form as in the LUBBY2 creep law described above is employed, but the six parameters are considered to be functions of the temperature (but not the stress, because the stress effects on parameters $\bar{\eta}_M, \bar{G}_K, \bar{\eta}_K$ have already been taken into account using the three parameters m, k_1, k_2). The six parameters can also be evaluated using the user-supplied subroutines mentioned above.

Irradiation creep model

The irradiation creep strain is calculated from

$${}^\phi \bar{\epsilon}^C = \left[1 - \exp(-a_1 \phi) \right] \frac{a_2 {}^\phi \sigma}{E_0} + M {}^\phi \sigma \phi,$$

$$M = a_3 \exp(a_4 \theta + a_5)$$

where ϕ is the neutron fluence at time t , a_1, \dots, a_5 are material constants, θ is the Kelvin temperature at time t , and E_0 is the initial Young's modulus. The subsequent Young's modulus can be dependent on neutron fluence and temperature. In this case, its

value is determined by using the Shifting rule described in Section 3.6.5.

All creep material models

- The creep strains are evaluated using the strain hardening procedure for load and temperature variations, and the O.R.N.L. rules for cyclic loading conditions.

ref. C.E. Pugh, J.M. Corum, K.C. Liu and W.L. Greenstreet, "Currently Recommended Constitutive Equations for Inelastic Design of FFTF Components," *Report No. TM-3602, Oak Ridge National Laboratory, Oak Ridge, Tennessee, 1972.*

The procedure used to evaluate the incremental creep strains is summarized in the following: Given the total creep strains ${}^t e_{ij}^C$ and the deviatoric stresses ${}^{t+\Delta t} s_{ij}$,

- 1) Calculate the effective stress

$${}^{t+\Delta t} \bar{\sigma} = \left[\frac{3}{2} {}^{t+\Delta t} s_{ij} {}^{t+\Delta t} s_{ij} \right]^{\frac{1}{2}}$$

- 2) Calculate the pseudo-effective creep strain

$${}^t \bar{e}^C = \left[\frac{2}{3} \left({}^t \bar{e}_{ij}^C - e_{ij}^{orig} \right) \left({}^t \bar{e}_{ij}^C - e_{ij}^{orig} \right) \right]^{\frac{1}{2}}$$

- 3a) For creep law 1 (and also for creep law 3 with constant coefficients when the constants correspond to creep law 1), calculate the effective creep strain and effective creep strain rate at time $t + \Delta t$ using

$$\left({}^{t+\Delta t} \bar{e}^C \right)^{1/a_2} = \left({}^t \bar{e}^C \right)^{1/a_2} + \left(a_0 {}^{t+\Delta t} \bar{\sigma}^{a_1} \right)^{1/a_2} \Delta t ,$$

$${}^{t+\Delta t}\dot{\bar{e}}^C = \frac{{}^{t+\Delta t}\bar{e}^C - {}^t\bar{e}^C}{\Delta t}$$

3b) For other creep laws (including creep law 3 with variable coefficients), calculate the pseudo-time \bar{t} satisfying

$$\bar{e}^C({}^{t+\Delta t}\bar{\sigma}, {}^{t+\Delta t}\theta, \bar{t}) = {}^t\bar{e}^C + \dot{\bar{e}}^C({}^{t+\Delta t}\bar{\sigma}, {}^{t+\Delta t}\theta, \bar{t})\Delta t$$

where $\bar{e}^C({}^{t+\Delta t}\bar{\sigma}, {}^{t+\Delta t}\theta, \bar{t})$ is the generalized uniaxial creep law

$$\text{and } \dot{\bar{e}}^C({}^{t+\Delta t}\bar{\sigma}, {}^{t+\Delta t}\theta, \bar{t}) = \frac{d\bar{e}^C({}^{t+\Delta t}\bar{\sigma}, {}^{t+\Delta t}\theta, \bar{t})}{d\bar{t}}. \text{ Then}$$

calculate the effective creep strain and effective creep strain rate at time $t + \Delta t$ using

$${}^{t+\Delta t}\bar{e}^C = \bar{e}^C({}^{t+\Delta t}\bar{\sigma}, {}^{t+\Delta t}\theta, \bar{t}), \quad {}^{t+\Delta t}\dot{\bar{e}}^C = \dot{\bar{e}}^C({}^{t+\Delta t}\bar{\sigma}, {}^{t+\Delta t}\theta, \bar{t}).$$

4) Calculate ${}^{t+\Delta t}\gamma$ using

$${}^{t+\Delta t}\gamma = \frac{3}{2} \frac{{}^{t+\Delta t}\dot{\bar{e}}^C}{{}^{t+\Delta t}\bar{\sigma}}$$

5) Calculate the incremental creep strains using

$$\Delta\bar{e}_{ij}^C = \Delta t \quad {}^{t+\Delta t}\gamma \quad {}^{t+\Delta t}s_{ij}$$

The use of the pseudo-time in step 3b corresponds to a strain hardening procedure. There is also the option of using time hardening in step 3b, in which $\bar{t} = t$. See ref. KJB, pp 607-608 for a discussion of the two hardening procedures. The strain hardening procedure is recommended for general use (and is the default).

The above procedure also assumes that the time integration parameter α is equal to 1 (the default). There is also the option of using a different value of α .

- Rupture of the material due to excessive creep strain can be specified. In this case, a uniaxial creep strain failure envelope must be defined as a function of temperature and effective stress (Fig. 3.6-5).

The following criterion is then checked at each integration point during the analysis:

$${}^t\bar{\epsilon}^C \leq \frac{e_{u,r}^C}{TF}$$

in which $e_{u,r}^C$ is the uniaxial creep strain at failure, ${}^t\bar{\epsilon}^C$ is the effective creep strain and TF is the triaxiality factor.

You can choose to use the triaxiality factor $TF = \frac{|{}^t\sigma_m|}{{}^t\bar{\sigma}}$ (the default) or to use no triaxiality factor (that is, $TF = 1$).

${}^t\sigma_m = {}^t\sigma_{11} + {}^t\sigma_{22} + {}^t\sigma_{33}$ is the volumetric stress.

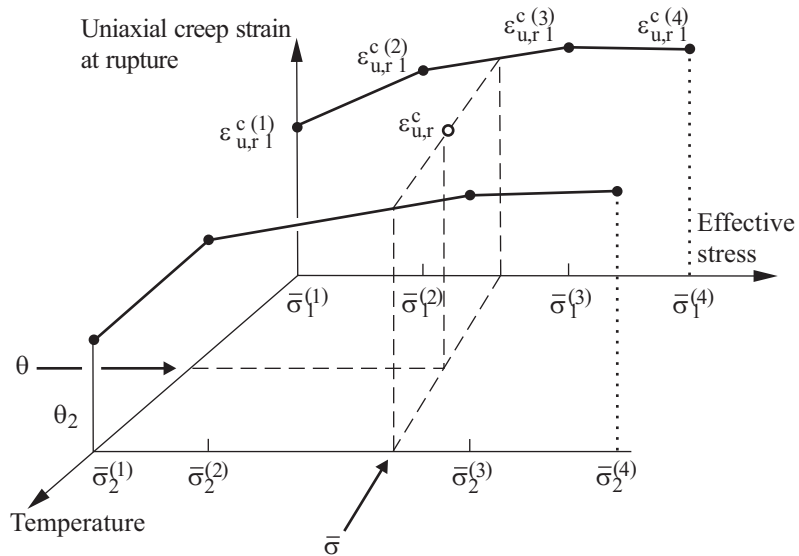


Figure 3.6-5: Multilinear interpolation of uniaxial creep strain

at rupture

You can choose to apply the rupture model only under tensile conditions (${}^t\sigma_m > 0$) or under both tensile and compressive conditions. The default is to apply the rupture model only under tensile conditions.

Rupture can be allowed only in tension, based on the sign of ${}^t\sigma_m$.

- The accumulated effective creep strain is calculated as

$${}^t\bar{\epsilon}^C = \sum_{\text{all solution steps}} \Delta\bar{\epsilon}^C$$

where $\Delta\bar{\epsilon}^C = \sqrt{\frac{2}{3} \Delta\mathbf{e}^C \cdot \Delta\mathbf{e}^C}$ and $\Delta\mathbf{e}^C$ is the tensor of creep strain increments in a solution step. Because of the summation over the solution steps, we refer to the calculated value of ${}^t\bar{\epsilon}^C$ as the accumulated effective creep strain.

3.6.1.4 Computational procedures

ref. KJB
Section 6.6.3

- The stresses at the integration points are evaluated using the effective-stress-function algorithm.

ref. M. Kojić and K.J. Bathe, "The Effective-Stress-Function Algorithm for Thermo-Elasto-Plasticity and Creep," *Int. J. Numer. Meth. Engng.*, Vol. 24, No. 8, pp. 1509-1532, 1987.

Briefly, the procedure used consists of the following calculations. The general constitutive equation

$${}^{t+\Delta t}\boldsymbol{\sigma}^{(i)} = {}^{t+\Delta t}\mathbf{C}^E \left({}^{t+\Delta t}\mathbf{e}^{(i)} - {}^{t+\Delta t}\mathbf{e}^{P(i)} - {}^{t+\Delta t}\mathbf{e}^{C(i)} - {}^{t+\Delta t}\mathbf{e}^{TH} \right) \quad (3.6-1)$$

is solved separately for the mean stress and for the deviatoric stresses. In this equation the index (i) denotes the iteration counter in the iteration for nodal point equilibrium. For easier writing this index will be dropped in the discussion to follow. The mean stress is calculated as

$${}^{t+\Delta t}\sigma_m = \frac{{}^{t+\Delta t}E}{1-2{}^{t+\Delta t}\nu} \left({}^{t+\Delta t}e_m - {}^{t+\Delta t}e^{TH} \right) \quad (3.6-2)$$

The deviatoric stresses ${}^{t+\Delta t}\mathbf{s}$ depend on the inelastic strains and they can be expressed as

$${}^{t+\Delta t}\mathbf{s} = \frac{1}{{}^{t+\Delta t}a_E + \alpha \Delta t {}^\tau\gamma + \Delta\lambda} \left[{}^{t+\Delta t}\mathbf{e}'' - (1-\alpha)\Delta t {}^\tau\gamma {}^t\mathbf{s} \right] \quad (3.6-3)$$

where ${}^{t+\Delta t}a_E = \frac{{}^{t+\Delta t}E}{1+{}^{t+\Delta t}\nu}$, ${}^t\mathbf{s}$ = deviatoric stress at the start of the time step and α is the integration parameter used for stress evaluation ($0 \leq \alpha < 1$) The creep and plastic multipliers ${}^\tau\gamma$ and $\Delta\lambda$ are functions of the effective stress ${}^{t+\Delta t}\bar{\sigma}$ only, and they account for creep and plasticity; also

$${}^{t+\Delta t}\mathbf{e}'' = {}^{t+\Delta t}\mathbf{e}' - {}^t\mathbf{e}^P - {}^t\mathbf{e}^C$$

is known since the deviatoric strains ${}^{t+\Delta t}\mathbf{e}'$, plastic strains ${}^t\mathbf{e}^P$ and creep strains ${}^t\mathbf{e}^C$ are known from the current displacements and the stress/strain state at the start of the current time step.

The following scalar function $f({}^{t+\Delta t}\bar{\sigma})$ is obtained from Eq. (3.6-3)

$$f({}^{t+\Delta t}\bar{\sigma}) = a^2 {}^{t+\Delta t}\bar{\sigma}^2 + b {}^\tau\gamma - c^2 {}^\tau\gamma^2 - d^2 = 0 \quad (3.6-4)$$

The zero of equation (3.6-4) provides the solution for the effective stress ${}^{t+\Delta t}\bar{\sigma}$, where

$$a = {}^{t+\Delta t}a_E + \alpha \Delta t {}^\tau\gamma + \Delta\lambda$$

$$\begin{aligned}
b &= 3(1-\alpha)\Delta t {}^{t+\Delta t}e'_{ij} {}^t s_{ij} \\
c &= (1-\alpha)\Delta t {}^t \bar{\sigma} \\
d^2 &= \frac{3}{2} {}^{t+\Delta t}e''_{ij} {}^{t+\Delta t}e''_{ij}
\end{aligned}$$

with summation on the indices i, j .

Once the solution for ${}^{t+\Delta t}\bar{\sigma}$ has been determined from Eq. (3.6-4), simultaneously with the scalars ${}^t\gamma$ and $\Delta\lambda$ from the creep and plasticity conditions, the deviatoric stress ${}^{t+\Delta t}\mathbf{s}$ is calculated from Eq. (3.6-3), and the plastic and creep strains at the end of the time step are obtained as

$$\begin{aligned}
{}^{t+\Delta t}\mathbf{e}^P &= {}^t\mathbf{e}^P + \Delta\lambda {}^{t+\Delta t}\mathbf{s} \\
{}^{t+\Delta t}\mathbf{e}^C &= {}^t\mathbf{e}^C + \left[(1-\alpha) {}^t\mathbf{s} + \alpha {}^{t+\Delta t}\mathbf{s} \right] \Delta t {}^t\gamma
\end{aligned}$$

The above equations correspond to isotropic hardening conditions and a general 3-D analysis. The solution details for kinematic hardening conditions and for special problems (for the plane stress, shell, isoparametric beam elements) are given in the above cited reference and also in the following reference.

ref. M. Kojić and K.J. Bathe, "Thermo-Elastic-Plastic and Creep Analysis of Shell Structures", *Computers & Structures*, Vol. 26, No 1/2, pp. 135-143, 1987.

3.6.1.5 Shifting rule for the irradiation creep model

- Irradiation tests obtaining irradiation strain data and material property changes of graphite have been carried out generally at a constant temperature. They cannot be determined directly from the test data due to variation of temperature.

Therefore, the irradiation strain at temperature θ and under fast neutron fluence ϕ is calculated as follows

$$\varepsilon^W(\theta, \phi) = \sum_{i=1}^n \Delta \varepsilon^W(\theta_i, \phi_{i-1}, \phi_i - \phi_{i-1})$$

where

$$\Delta \varepsilon^W(\theta_i, \phi, \Delta \phi) = \varepsilon^W(\theta_i, \phi + \Delta \phi) - \varepsilon^W(\theta_i, \phi)$$

This is illustrated in Figure 3.6-6, and documented in the following reference.

ref. T. Iyoku and M. Ishihara, "Development of Thermal/Irradiation Stress Analytical Code VIENUS for HTTR Graphite Block", *Journal of Nuclear Science and Technology*, 28[10], pp. 921-931, October 1991.

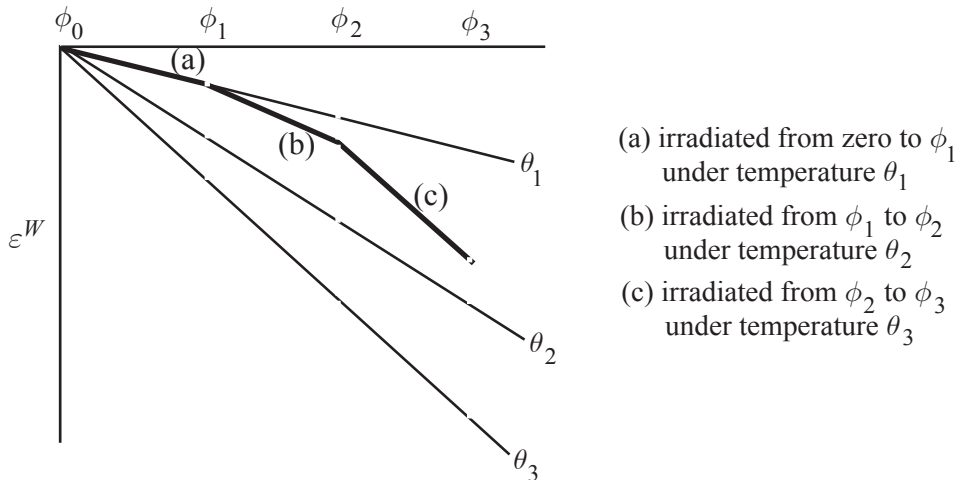


Figure 3.6-6: Shifting rule for irradiation creep model

3.6.2 Thermo-plastic-cyclic material model

- This section describes the **thermo-plastic-cyclic** material

model.

ref. KJB
Section 6.6.3

- The thermo-plastic-cyclic material model is an extension of the plastic-cyclic material model described in Section 3.4.5.

- This material model can be used with the **truss, 2-D solid, 3-D solid, beam, shell** and **3-D shell** elements.

- This model can be used with the **small displacement/small strain, large displacement/small strain** and **large displacement/large strain** formulations.

When used with the small displacement/small strain formulation, a materially-nonlinear-only formulation is employed, when used with the large displacement/small strain formulation, a TL is employed, and when used with the large displacement/large strain formulation, either a ULH or ULJ formulation is employed.

- If geometrically nonlinear effects are to be included, the large displacements/large strain kinematics are preferred to the large displacement/small strain kinematics, even when the strains are numerically small. By default large strain kinematics are used for 2-D solid and 3-D solid elements when large displacement kinematics are selected.

- Elements that reference this material model should also employ the mixed displacement-pressure (u/p) element formulation. The u/p formulation is the default for these elements.

3.6.2.1 Fundamental concepts

- Many of the ideas used in the thermo-plastic-cyclic material model are given in the following references:

ref J. Lemaitre and J.-L. Chaboche, *Mechanics of Solid Materials*, Cambridge University Press, 1990.

ref J.-L. Chaboche, "Cyclic Viscoplastic Constitutive Equations, part I: A Thermodynamically Consistent Formulation", *J. Appl. Mech.*, December 1993, Vol 60, pp 813-821.

The theory employed in the thermo-plastic-cyclic model is the same (with some exceptions given below) as the theory employed in the plastic-cyclic model. Therefore the descriptions below refer to the corresponding descriptions for the plastic-cyclic model (Section 3.4.5.1). And in the following, we discuss only the differences between the plastic-cyclic model and the thermo-plastic-cyclic model.

Interpolation of material constants as a function of

temperature: In the thermo-plastic-cyclic model, all material constants are functions of temperature. The material constants are entered for several temperatures as described in detail below. During the solution process, the current temperature at an integration point is determined. Then the material constants are interpolated to the current temperature according to the following process:

Let ${}^{t+\Delta t}\theta$ be the current temperature, and θ_i and θ_{i+1} be two successive user-input temperatures that bracket ${}^{t+\Delta t}\theta$. A generic material constant, for example the Young's modulus E , is specified at the two user-input temperatures: $E(\theta_i)$, $E(\theta_{i+1})$, and these specified values are interpolated to the current temperature using

$$E({}^{t+\Delta t}\theta) = E(\theta_i) + (E(\theta_{i+1}) - E(\theta_i)) \frac{{}^{t+\Delta t}\theta - \theta_i}{\theta_{i+1} - \theta_i}$$

Stresses, strains, stress-strain law: The stresses, strains and stress-strain law for the thermo-plastic-cyclic model are the same as for the plastic-cyclic model. However in the thermo-plastic-cyclic model, the material constants E , ν are functions of temperature.

Yield condition: The same von Mises yield condition is used in the thermo-plastic-cyclic and plastic-cyclic models.

Flow rule: The same flow rule is used in the thermo-plastic-cyclic and plastic cyclic models, with the exception that constant β of the plastic-cyclic model is replaced by 1.0 in the thermo-plastic-cyclic model. Therefore the flow rule of the thermo-plastic-cyclic

model is

directions of $\Delta \mathbf{e}^P$ = directions of $\left({}^{t+\Delta t} \mathbf{s} - {}^{t+\Delta t} \mathbf{a} \right)$

Strain memory surface: The same strain memory surface concept is used in the thermo-plastic-cyclic and plastic-cyclic models, with the exception that constant β of the plastic-cyclic model is replaced by 1.0 in the thermo-plastic-cyclic model.

Isotropic hardening rules: The same isotropic hardening rules are used in the thermo-plastic-cyclic and plastic-cyclic models, with the exceptions and modifications given below.

Bilinear hardening:

$${}^{t+\Delta t} \sigma_y = {}^0 \sigma_y + E_p {}^{t+\Delta t} \bar{\epsilon}^P$$

where ${}^0 \sigma_y$ and E_p are material constants. In the thermo-plastic-cyclic model, ${}^0 \sigma_y$ and E_p are functions of temperature, and are interpolated to the current temperature as described above.

Notice that if the temperature suddenly changes, it is possible that material constants ${}^0 \sigma_y$ and E_p suddenly change, and therefore that the yield stress also suddenly changes.

Multilinear hardening:

For each user-specified temperature, pairs of $(\bar{\epsilon}^P, \sigma_y)$ are given. It is allowed to have a different number of pairs for each specified temperature.

Then, for the current temperature ${}^{t+\Delta t} \theta$ and current accumulated effective plastic strain ${}^{t+\Delta t} \bar{\epsilon}^P$, the current yield stress ${}^{t+\Delta t} \sigma_y$ is determined by interpolation, as shown in Fig. 3.6-3.

As in the case of bilinear hardening, if the temperature suddenly

changes, the yield stress can also suddenly change.

Exponential hardening:

Because the material constants can be functions of temperature, the procedure used in the thermo-plastic-cyclic model is different than the procedure used in the plastic-cyclic model. Here are the details.

In the plastic-cyclic model, the equation

$${}^{t+\Delta t}\sigma_y = {}^0\sigma_y + Q(1 - \exp(-b {}^{t+\Delta t}\bar{\epsilon}^P))$$

is used, in which Q and b are material constants. However, in the thermo-plastic-cyclic model, Q and b are functions of temperature, interpolated to the current temperature as described above.

The above equation assumes isothermal conditions, so the above equation gives nonphysical results under varying temperature conditions. Therefore, following the 1993 paper by Chaboche, we use the following expressions in the thermo-plastic-cyclic model:

$$\begin{aligned}\sigma_y &= {}^0\sigma_y + R \\ R &= bQr \\ dr &= (1 - br)d\bar{\epsilon}^P\end{aligned}$$

(this is the “second option” presented in Section 4.2 of the 1993 paper). The left superscript $t + \Delta t$ is omitted from these expressions, but is implied.

Within a time step, the differential equation

$$dr = (1 - br)d\bar{\epsilon}^P$$

is analytically integrated using the current value of b . This procedure gives the same results as the plastic-cyclic model when the thermo-plastic-cyclic model is used under isothermal

conditions.

Memory-exponential hardening (exponential hardening with strain memory surface):

Because the material constants can be functions of temperature, the procedure used in the thermo-plastic-cyclic model is different than the procedure used in the plastic-cyclic model. Here are the details.

In the plastic-cyclic model, we use

$$\begin{aligned}\sigma_y &= {}^0\sigma_y + R \\ dR &= b(Q - R)d\bar{\epsilon}^P \\ Q &= Q_M - (Q_M - Q_0)\exp(-2\mu q)\end{aligned}$$

in which ${}^0\sigma_y$, b , Q_M , Q_0 , μ are material properties.

Unfortunately we cannot use $dR = b(Q - R)d\bar{\epsilon}^P$ in the thermo-plastic-cyclic model because $dR = b(Q - R)d\bar{\epsilon}^P$ applies only to isothermal conditions, so we use, instead,

$$\begin{aligned}\sigma_y &= {}^0\sigma_y + R \\ R &= bQ_D r \\ \tilde{q} &= \frac{Q_M - (Q_M - Q_0)\exp(-2\mu q)}{Q_D} \\ dr &= (\tilde{q} - br)d\bar{\epsilon}^P\end{aligned}$$

$$Q_D = \max(|Q_0|, |Q_M|, |Q_M - Q_0|)$$

in which ${}^0\sigma_y$, b , Q_M , Q_0 , μ are functions of temperature. This procedure is somewhat arbitrary, but has the following characteristics.

- 1) This procedure reduces to the procedure used in the plastic-

cyclic model when the thermo-plastic-cyclic model is used under isothermal conditions.

2) If $Q_M = Q_0$, then $\tilde{q} = 1$ and this procedure reduces to the exponential hardening procedure given above.

2) When Q_M, Q_0 are temperature-dependent such that their ratio is temperature-independent, then \tilde{q} is temperature-independent, too.

Within a time step, the differential equation

$$dr = (\tilde{q} - br) d\bar{e}^P$$

is analytically integrated using the current values of b and \tilde{q} .

This integration is different than the integration used in the plastic-cyclic model. In the plastic-cyclic model, the differential equation

$$dR = b(Q - R) d\bar{e}^P$$

is numerically integrated using a backwards difference method:

${}^{t+\Delta t}R = {}^tR + \Delta R$, $\Delta R = b({}^{t+\Delta t}Q - {}^{t+\Delta t}R)\Delta\bar{e}^P$. Therefore the numerical results from the thermo-plastic-cyclic and plastic-cyclic models will be different, even when the thermo-plastic-cyclic model is used under isothermal conditions. However the differences will be small when the time step size is small.

Kinematic hardening rule: The kinematic hardening rule is:

Armstrong-Fredrick nonlinear kinematic hardening:

In both the plastic-cyclic and thermo-plastic-cyclic models, the back stress is expressed as a sum of partial back stresses

$${}^{t+\Delta t}\boldsymbol{\alpha} = \sum {}^{t+\Delta t}\boldsymbol{\alpha}^{(m)}$$

where ${}^{t+\Delta t}\boldsymbol{\alpha}^{(m)}$ is partial back stress number m . All of the partial back stresses are independent of each other.

In the plastic-cyclic model, each partial back stress evolves according to the differential rule

$$d\boldsymbol{\alpha}^{(m)} = \frac{2}{3}h^{(m)}d\mathbf{e}^P - \zeta^{(m)} {}^t\boldsymbol{\alpha}^{(m)} d\bar{e}^P$$

where $h^{(m)}$ and $\zeta^{(m)}$ are material constants. However, this procedure gives nonphysical results when the material constants are functions of temperature. Instead, for the thermo-plastic-cyclic model, the procedure described in the 1993 paper by Chaboche is used. Each partial back stress is calculated from

$${}^{t+\Delta t}\boldsymbol{\alpha}^{(m)} = \frac{2}{3}h^{(m)} {}^{t+\Delta t}\mathbf{a}^{(m)}$$

where ${}^{t+\Delta t}\mathbf{a}^{(m)}$ is a quantity that is thermodynamically conjugate to ${}^{t+\Delta t}\boldsymbol{\alpha}^{(m)}$. ${}^{t+\Delta t}\mathbf{a}^{(m)}$ evolves according to the differential rule

$$d\mathbf{a}^{(m)} = d\mathbf{e}^P - \zeta^{(m)} {}^t\mathbf{a}^{(m)} d\bar{e}^P$$

Initially $\mathbf{a}^{(m)} = 0$.

Carefully notice that $h^{(m)}$ and $\boldsymbol{\alpha}^{(m)}$ can instantaneously change when the temperature changes, however $\mathbf{a}^{(m)}$ cannot. This issue is discussed more fully in the 1993 paper by Chaboche.

The so-called back stress temperature correction described in Section 3.6.2 is automatically included in the thermo-plastic-cyclic model.

Under isothermal conditions, the procedures used for the plastic-cyclic and thermo-plastic-cyclic models give the same results.

$h^{(m)}$ and $\zeta^{(m)}$ have the same meanings for both the plastic-cyclic

and thermo-plastic-cyclic models, so the discussion of the stress-plastic strain curve for uniaxial cycling and the stress-plastic strain curve for initial loading, given in Section 3.4.5.1, applies directly to the thermo-plastic-cyclic model.

Combination of isotropic and kinematic hardening rules: The isotropic and kinematic hardening rules can be combined in the thermo-plastic-cyclic model, exactly as in the plastic-cyclic model.

Stress-strain integration: When plasticity is detected, the incremental plastic strains are solved for using an iterative solution procedure, as in the plastic-cyclic model. The maximum number of iterations can be specified, and the tolerance used in assessing convergence can be specified.

Constitutive tensor: As in the plastic-cyclic model, the constitutive tensor (stress-strain matrix) is constructed to be tangent. This gives good convergence in the equilibrium iterations. However, in general, the tangent constitutive tensor is non-symmetric. Therefore it is allowed to use the non-symmetric equation solver when the thermo-plastic-cyclic model is used. It is also allowed to use a symmetric equation solver; in this case, the constitutive tensor is symmetrized, and this will reduce the rate of convergence (increase the number of equilibrium iterations).

Formulations: Exactly the same formulations (small displacement/small strain, large displacement/small strain, large displacement/large strain) are available for the thermo-plastic-cyclic model as for the plastic-cyclic model.

Mixed displacement-pressure formulation: As in the plastic-cyclic model, elements that reference these material models should also employ the mixed displacement-pressure (u/p) element formulation.

Thermal strains: If a thermal load is applied to the structure, the thermal strains are taken into account. In the thermo-plastic-cyclic model, the coefficient of thermal expansion is temperature dependent. The procedure used is the same as for the thermo-isotropic material model, see Section 3.5.

The thermal strains are always isotropic.

Modeling of rupture: The same rupture criteria available for the plastic-cyclic model are available for the thermo-plastic-cyclic model. The maximum accumulated effective plastic strain can be specified, and this maximum accumulated effective plastic strain can be a function of temperature.

3.6.2.2 Specification of input

MATERIAL THERMO-PLASTIC-CYCLIC: The basic command is

```
MATERIAL THERMO-PLASTIC-CYCLIC NAME,  
          TREF DENSITY MAXITE RTOL  
thetai Ei nui alphai plcycl-isotropici ,  
                               plcycl-kinematici ,  
                               plcycl-rupturei
```

This command references data defined in commands *PLCYCL-ISOTROPIC*, *PLCYCL-KINEMATIC*, *PLCYCL-RUPTURE*. These commands provide the input for the isotropic, kinematic and rupture parts of the thermo-plastic-cyclic model. Each of the isotropic, kinematic and rupture options given above has a corresponding *PLCYCL-ISOTROPIC*, *PLCYCL-KINEMATIC* and *PLCYCL-RUPTURE* command, see Section 3.4.5.2 for a detailed description of the options.

plcycl-isotropic_i must always be specified (non-zero).

plcycl-kinematic_i and *plcycl-rupture_i* can either be zero or non-zero. Zero means that the effect is not included.

The following restrictions apply:

- 1) There must be at least two temperature rows in the *MATERIAL THERMO-PLASTIC-CYCLIC* command input. There is no limit on the number of temperature rows.
- 2) All of the isotropic hardening definitions specified must be of the same type. For all example, all of the isotropic hardening

definitions can be specified using PLCYCL-ISOTROPIC BILINEAR commands. However it is an error to specify a PLCYCL-ISOTROPIC BILINEAR definition and a PLCYCL-ISOTROPIC MULTILINEAR definition.

Example showing valid input for PLCYCL-ISOTROPIC:

```
PLCYCL-ISOTROPIC BILINEAR 1 ...
PLCYCL-ISOTROPIC BILINEAR 2 ...
MATERIAL THERMO-PLASTIC-CYCLIC 1 ...
ENTRIES THETA    ... PLCYCL-ISOTROPIC
           0.0    ...                      1
           100.0  ...                      2
...

```

Example showing invalid input for PLCYCL-ISOTROPIC:

```
PLCYCL-ISOTROPIC BILINEAR 1 ...
PLCYCL-ISOTROPIC MULTILINEAR 2 ...
MATERIAL THERMO-PLASTIC-CYCLIC 1 ...
ENTRIES THETA    ... PLCYCL-ISOTROPIC
           0.0    ...                      1
           100.0  ...                      2
...

```

3) All of the kinematic hardening definitions specified must be of the same type. For all example, all of the kinematic hardening definitions can be specified using the PLCYCL-KINEMATIC ARMSTRONG-FREDRICK command. However it is an error to specify a PLCYCL-KINEMATIC ARMSTRONG-FREDRICK definition and a zero PLCYCL-KINEMATIC definition.

Also, if the kinematic hardening definitions are all of type ARMSTRONG-FREDRICK, each definition must have the same number of rows in the Armstrong-Fredrick definition.

Example showing valid input for PLCYCL-KINEMATIC:

```
PLCYCL-KINEMATIC ARMSTRONG-FREDRICK 1 ...
100.0 0.1
200.0 0.05
PLCYCL-KINEMATIC ARMSTRONG-FREDRICK 2 ...
50.0 0.01
70.0 0.02

```



```
MATERIAL THERMO-PLASTIC-CYCLIC 1 ...
ENTRIES THETA    ... PLCYCL-KINEMATIC
          0.0    ...                      1
          100.0  ...                      2
...

```

Example 1 showing invalid input for PLCYCL-KINEMATIC:

```
PLCYCL-KINEMATIC ARMSTRONG-FREDRICK 1 ...
100.0 0.1
200.0 0.05
MATERIAL THERMO-PLASTIC-CYCLIC 1 ...
ENTRIES THETA    ... PLCYCL-KINEMATIC
          0.0    ...                      1
          100.0  ...                      0
...

```

Example 2 showing invalid input for PLCYCL-KINEMATIC:

```
PLCYCL-KINEMATIC ARMSTRONG-FREDRICK 1 ...
100.0 0.1
200.0 0.05
PLCYCL-KINEMATIC ARMSTRONG-FREDRICK 2 ...
50.0 0.01
70.0 0.02
100.0 0.03
MATERIAL THERMO-PLASTIC-CYCLIC 1 ...
ENTRIES THETA    ... PLCYCL-KINEMATIC
          0.0    ...                      1
          100.0  ...                      2
...

```

4) All of the rupture definitions specified must be of the same type. For all example, all of the rupture definitions can be specified using the PLCYCL-RUPTURE AEPS command. However it is an error to specify a PLCYCL-RUTPURE AEPS definition and a zero PLCYCL-RUPTURE definition.

Example showing valid input for PLCYCL-RUPTURE:

```
PLCYCL-RUPTURE AEPS 1 ...
PLCYCL-RUPTURE AEPS 2 ...
MATERIAL THERMO-PLASTIC-CYCLIC 1 ...
ENTRIES THETA    ... PLCYCL-RUPTURE

```

```

          0.0    ...          1
        100.0    ...          2
...

```

Example showing invalid input for PLCYCL-RUPTURE:

```

PLCYCL-RUPTURE AEPS 1 ...
MATERIAL THERMO-PLASTIC-CYCLIC 1 ...
ENTRIES THETA    ... PLCYCL-RUPTURE
          0.0    ...          1
        100.0    ...          0
...

```

3.6.2.3 Output variables

The thermo-plastic-cyclic material model outputs the same information as the plastic-cyclic model: stresses, strains, plastic strains, thermal strains, accumulated effective plastic strain ${}^{t+\Delta t}\bar{\epsilon}^P$,

yield stress ${}^{t+\Delta t}\sigma_y$, stress radius (defined as $\sqrt{\frac{3}{2}} \| {}^{t+\Delta t}\mathbf{s} - {}^{t+\Delta t}\mathbf{a} \|$)

and dissipated plastic work (defined as $w_p = \int_0^{t+\Delta t} \boldsymbol{\tau} : d\mathbf{e}^P$). When

the strain memory surface is used, the strain memory size ${}^{t+\Delta t}q$ is also output.

3.6.3 Evaluation of creep strains

Creep, plastic-creep, multilinear-plastic-creep material models

- The effective creep strain is calculated using one of the following creep laws:

Power creep law (creep law 1) :

$${}^t\bar{\epsilon}^C = a_0 {}^t\sigma^{a_1} t^{a_2}$$

in which a_0, a_1, a_2 are material constants.

Exponential creep law (creep law 2) :

$${}^t\bar{\epsilon}^C = F(1 - e^{-Rt}) + Gt$$

with

$$F = a_0 e^{a_1 {}^t\sigma}; \quad R = a_2 \left(\frac{{}^t\sigma}{a_3} \right)^{a_4}; \quad G = a_5 e^{a_6 {}^t\sigma}$$

in which a_0 to a_6 are material constants.

Eight-parameter creep law (creep law 3) :

$${}^t\bar{\epsilon}^C = S \cdot T \cdot e^{-H}$$

with

$$S = a_0 {}^t\sigma^{a_1}; \quad T = t^{a_2} + a_3 t^{a_4} + a_5 t^{a_6}, \quad H = \frac{a_7}{t\theta}$$

in which a_0 to a_7 are material constants.

In the above equations, ${}^t\bar{\epsilon}^C$, ${}^t\sigma$ and ${}^t\theta$ denote the effective creep strain, effective stress and temperature at time t . The unit of the temperature can either be K or °C (default). ADINA automatically applies the conversion to K in the default setting.

LUBBY2 creep law: The LUBBY2 creep law can be employed to predict long-term behavior of rock salt mass.

The LUBBY2 creep law is implemented for 2-D solid and 3-D solid elements. It can be used in conjunction with strain-hardening or time-hardening.

The LUBBY2 creep law is specified by the following equations:

$$\frac{d^t\bar{e}^C}{dt} = \left[\frac{1}{\bar{\eta}_K(^t\bar{\sigma})} \exp\left(-\frac{\bar{G}_K(^t\bar{\sigma})}{\bar{\eta}_K(^t\bar{\sigma})}t\right) + \frac{1}{\bar{\eta}_M(^t\bar{\sigma})} \right] ^t\bar{\sigma}$$

or

$$^t\bar{e}^C = \left[\frac{-1}{\bar{G}_K(^t\bar{\sigma})} \exp\left(-\frac{\bar{G}_K(^t\bar{\sigma})}{\bar{\eta}_K(^t\bar{\sigma})}t\right) + \frac{t}{\bar{\eta}_M(^t\bar{\sigma})} + \frac{1}{\bar{G}_K(^t\bar{\sigma})} \right] ^t\bar{\sigma}$$

where $\bar{\eta}_M(^t\bar{\sigma}) = \bar{\eta}_M^* e^{m^t\bar{\sigma}}$, $\bar{G}_K(^t\bar{\sigma}) = \bar{G}_K^* e^{k_1^t\bar{\sigma}}$ and $\bar{\eta}_K(^t\bar{\sigma}) = \bar{\eta}_K^* e^{k_2^t\bar{\sigma}}$. Also, as usual, $^t\bar{e}^C$ is the effective creep strain, $\frac{d^t\bar{e}^C}{dt}$ is the effective creep strain rate and $^t\bar{\sigma}$ is the effective creep stress.

In these equations, m , k_1 , k_2 , $\bar{\eta}_M^*$, \bar{G}_K^* and $\bar{\eta}_K^*$ are six material parameters. m , k_1 , k_2 account for the stress effect on parameters $\bar{\eta}_M^*$, \bar{G}_K^* , $\bar{\eta}_K^*$. The six material parameters are identified in ADINA as a0, a1, a2, a3, a4 and a5 respectively.

Unloading and cyclic loading effects are considered in the LUBBY2 creep law.

Analytical solutions with the LUBBY2 creep law are very sensitive to the three parameters m , k_1 , k_2 , so these parameters must be very carefully specified.

For more information about the LUBBY2 creep law, see the following reference:

ref. K.H. Lux and S. Heusermann, "Creep Tests on Rock Salt with Changing Load as a Basis for the Verification of Theoretical Material Laws", *Proc. 6th Symp. on Salt*, Toronto, Vol. 1, pp. 417-435, 1983.

Blackburn creep law:

$${}^t\bar{\epsilon}^C = C_1 \{1 - \exp(-r_1 t)\} + C_2 \{1 - \exp(-r_2 t)\} + Gt$$

with

$$C_1 = \frac{A_0 G^{A_1}}{r_1}, \quad C_2 = \frac{A_2 G^{A_3}}{r_2}$$

$$r_1 = A_4 t_R^{-A_5}, \quad r_2 = A_6 t_R^{-A_7}, \quad G = F t_R^{-A_8}$$

$$\log_{10}(t_R) = B_0 + B_1 \log_{10} {}^t\sigma + B_2 \left(\log_{10} {}^t\sigma\right)^2 + A_9$$

$$F = A_{10} \exp\left(\frac{A_{11}}{{}^t\theta + 273.16}\right)$$

$$B_0 = \frac{A_{12}}{{}^t\theta + 273.16} + A_{13}, \quad B_1 = \frac{A_{14}}{{}^t\theta + 273.16}, \quad B_2 = \frac{A_{15}}{{}^t\theta + 273.16}$$

$A_0, A_1, A_2, \dots, A_{15}$ are material constants. The unit of temperature should be Celsius.

Creep-variable, plastic-creep-variable, multilinear-plastic-creep-variable material models

The following creep laws can be employed:

Eight-parameter creep law (creep law 3) with variable coefficients: The same functional form as in creep law 3 described above is employed, but coefficients a_0 to a_7 are considered to be functions of temperature and effective stress.

The values of a_0 to a_7 can also be evaluated within user-supplied subroutines. In the user-supplied subroutines, these parameters can be functions of the temperature and the effective stress.

The user-supplied subroutines are:

UCOEF2 for 2-D solid elements (file ovl30u.f)

UCOEF3 for 3-D solid elements (file ovl40u.f)

UCOEFB for iso-beam elements (file ovl60u.f)

UCOEFS for shell elements (file ovl100u.f)
UCOEFP for pipe elements (file ovl110u.f)

These subroutines are called by ADINA every time the stress-strain law is updated.

Six-parameter LUBBY2 creep law with variable coefficients: The same functional form as in the LUBBY2 creep law described above is employed, but the six parameters are considered to be functions of the temperature (but not the stress, because the stress effects on parameters $\bar{\eta}_M, \bar{G}_K, \bar{\eta}_K$ have already been taken into account using the three parameters m, k_1, k_2). The six parameters can also be evaluated using the user-supplied subroutines mentioned above.

Irradiation creep model

The irradiation creep strain is calculated from

$${}^{\phi}\bar{\epsilon}^C = \left[1 - \exp(-a_1\phi) \right] \frac{a_2 {}^{\phi}\sigma}{E_0} + M {}^{\phi}\sigma \phi,$$

$$M = a_3 \exp(a_4\theta + a_5)$$

where ϕ is the neutron fluence at time t , a_1, \dots, a_5 are material constants, θ is the Kelvin temperature at time t , and E_0 is the initial Young's modulus. The subsequent Young's modulus can be dependent on neutron fluence and temperature. In this case, its value is determined by using the Shifting rule described in Section 3.6.5.

All creep material models

- The creep strains are evaluated using the strain hardening procedure for load and temperature variations, and the O.R.N.L. rules for cyclic loading conditions.

ref. C.E. Pugh, J.M. Corum, K.C. Liu and W.L. Greenstreet,
"Currently Recommended Constitutive Equations for
Inelastic Design of FFTF Components," *Report No. TM-*

3602, Oak Ridge National Laboratory, Oak Ridge, Tennessee, 1972.

The procedure used to evaluate the incremental creep strains is summarized in the following: Given the total creep strains ${}^t e_{ij}^C$ and the deviatoric stresses ${}^{t+\Delta t} s_{ij}$,

- 1) Calculate the effective stress

$${}^{t+\Delta t} \bar{\sigma} = \left[\frac{3}{2} {}^{t+\Delta t} s_{ij} {}^{t+\Delta t} s_{ij} \right]^{\frac{1}{2}}$$

- 2) Calculate the pseudo-effective creep strain

$${}^t \bar{e}^C = \left[\frac{2}{3} \left({}^t \bar{e}_{ij}^C - e_{ij}^{orig} \right) \left({}^t \bar{e}_{ij}^C - e_{ij}^{orig} \right) \right]^{\frac{1}{2}}$$

- 3a) For creep law 1 (and also for creep law 3 with constant coefficients when the constants correspond to creep law 1), calculate the effective creep strain and effective creep strain rate at time $t + \Delta t$ using

$$\begin{aligned} \left({}^{t+\Delta t} \bar{e}^C \right)^{1/a_2} &= \left({}^t \bar{e}^C \right)^{1/a_2} + \left(a_0 {}^{t+\Delta t} \bar{\sigma}^{a_1} \right)^{1/a_2} \Delta t, \\ {}^{t+\Delta t} \dot{\bar{e}}^C &= \frac{{}^{t+\Delta t} \bar{e}^C - {}^t \bar{e}^C}{\Delta t} \end{aligned}$$

- 3b) For other creep laws (including creep law 3 with variable coefficients), calculate the pseudo-time \bar{t} satisfying

$$\bar{e}^C \left({}^{t+\Delta t} \bar{\sigma}, {}^{t+\Delta t} \theta, \bar{t} \right) = {}^t \bar{e}^C + \dot{\bar{e}}^C \left({}^{t+\Delta t} \bar{\sigma}, {}^{t+\Delta t} \theta, \bar{t} \right) \Delta t$$

where $\bar{e}^C \left({}^{t+\Delta t} \bar{\sigma}, {}^{t+\Delta t} \theta, \bar{t} \right)$ is the generalized uniaxial creep law

and $\dot{\bar{e}}^C \left({}^{t+\Delta t}\bar{\sigma}, {}^{t+\Delta t}\theta, \bar{t} \right) = \frac{d\dot{\bar{e}}^C \left({}^{t+\Delta t}\bar{\sigma}, {}^{t+\Delta t}\theta, \bar{t} \right)}{d\bar{t}}$. Then

calculate the effective creep strain and effective creep strain rate at time $t + \Delta t$ using

$${}^{t+\Delta t}\bar{e}^C = \bar{e}^C \left({}^{t+\Delta t}\bar{\sigma}, {}^{t+\Delta t}\theta, \bar{t} \right), \quad {}^{t+\Delta t}\dot{\bar{e}}^C = \dot{\bar{e}}^C \left({}^{t+\Delta t}\bar{\sigma}, {}^{t+\Delta t}\theta, \bar{t} \right).$$

4) Calculate ${}^{t+\Delta t}\gamma$ using

$${}^{t+\Delta t}\gamma = \frac{3}{2} \frac{{}^{t+\Delta t}\dot{\bar{e}}^C}{{}^{t+\Delta t}\bar{\sigma}}$$

5) Calculate the incremental creep strains using

$$\Delta \bar{e}_{ij}^C = \Delta t \quad {}^{t+\Delta t}\gamma \quad {}^{t+\Delta t}s_{ij}$$

The use of the pseudo-time in step 3b corresponds to a strain hardening procedure. There is also the option of using time hardening in step 3b, in which $\bar{t} = t$. See ref. KJB, pp 607-608 for a discussion of the two hardening procedures. The strain hardening procedure is recommended for general use (and is the default).

The above procedure also assumes that the time integration parameter α is equal to 1 (the default). There is also the option of using a different value of α .

- Rupture of the material due to excessive creep strain can be specified. In this case, a uniaxial creep strain failure envelope must be defined as a function of temperature and effective stress (Fig. 3.6-5).

The following criterion is then checked at each integration point during the analysis:

$${}^t\bar{e}^C \leq \frac{e_{u,r}^C}{TF}$$

in which $e_{u,r}^C$ is the uniaxial creep strain at failure, ${}^t\bar{e}^C$ is the effective creep strain and TF is the triaxiality factor.

You can choose to use the triaxiality factor $TF = \frac{|\sigma_m|}{\bar{\sigma}}$ (the default) or to use no triaxiality factor (that is, $TF = 1$).

$\sigma_m = \sigma_{11} + \sigma_{22} + \sigma_{33}$ is the volumetric stress.

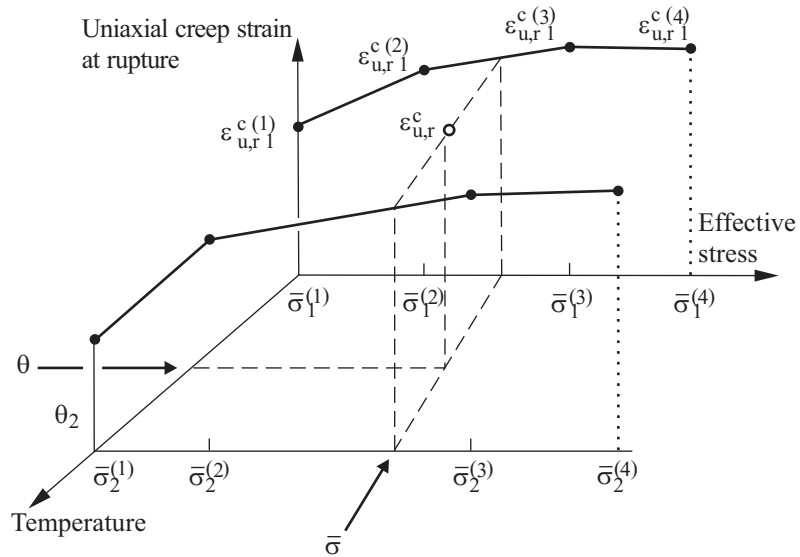


Figure 3.6-5: Multilinear interpolation of uniaxial creep strain at rupture

You can choose to apply the rupture model only under tensile conditions ($\sigma_m > 0$) or under both tensile and compressive conditions. The default is to apply the rupture model only under tensile conditions.

Rupture can be allowed only in tension, based on the sign of σ_m .

- The accumulated effective creep strain is calculated as

$$\epsilon^C = \sum_{\text{all solution steps}} \Delta \epsilon^C$$

where $\Delta \bar{\epsilon}^c = \sqrt{\frac{2}{3} \Delta \mathbf{e}^c \cdot \Delta \mathbf{e}^c}$ and $\Delta \mathbf{e}^c$ is the tensor of creep strain increments in a solution step. Because of the summation over the solution steps, we refer to the calculated value of $\bar{\epsilon}^c$ as the accumulated effective creep strain.

3.6.4 Computational procedures

ref. KJB
Section 6.6.3

- The stresses at the integration points are evaluated using the effective-stress-function algorithm.

ref. M. Kojić and K.J. Bathe, "The Effective-Stress-Function Algorithm for Thermo-Elasto-Plasticity and Creep," *Int. J. Numer. Meth. Engng.*, Vol. 24, No. 8, pp. 1509-1532, 1987.

Briefly, the procedure used consists of the following calculations. The general constitutive equation

$${}^{t+\Delta t} \boldsymbol{\sigma}^{(i)} = {}^{t+\Delta t} \mathbf{C}^E \left({}^{t+\Delta t} \mathbf{e}^{(i)} - {}^{t+\Delta t} \mathbf{e}^{P(i)} - {}^{t+\Delta t} \mathbf{e}^{C(i)} - {}^{t+\Delta t} \mathbf{e}^{TH} \right) \quad (3.6-1)$$

is solved separately for the mean stress and for the deviatoric stresses. In this equation the index (i) denotes the iteration counter in the iteration for nodal point equilibrium. For easier writing this index will be dropped in the discussion to follow. The mean stress is calculated as

$${}^{t+\Delta t} \sigma_m = \frac{{}^{t+\Delta t} E}{1 - 2 \nu} \left({}^{t+\Delta t} e_m - {}^{t+\Delta t} e^{TH} \right) \quad (3.6-2)$$

The deviatoric stresses ${}^{t+\Delta t} \mathbf{s}$ depend on the inelastic strains and they can be expressed as

$${}^{t+\Delta t} \mathbf{s} = \frac{1}{{}^{t+\Delta t} a_E + \alpha \Delta t^\tau \gamma + \Delta \lambda} \left[{}^{t+\Delta t} \mathbf{e}'' - (1 - \alpha) \Delta t^\tau \gamma {}^t \mathbf{s} \right] \quad (3.6-3)$$

where ${}^{t+\Delta t}a_E = \frac{{}^{t+\Delta t}E}{1 + {}^{t+\Delta t}\nu}$, ${}^t\mathbf{s}$ = deviatoric stress at the start of the time step and α is the integration parameter used for stress evaluation ($0 \leq \alpha < 1$) The creep and plastic multipliers ${}^\tau\gamma$ and $\Delta\lambda$ are functions of the effective stress ${}^{t+\Delta t}\bar{\sigma}$ only, and they account for creep and plasticity; also

$${}^{t+\Delta t}\mathbf{e}'' = {}^{t+\Delta t}\mathbf{e}' - {}^t\mathbf{e}^P - {}^t\mathbf{e}^C$$

is known since the deviatoric strains ${}^{t+\Delta t}\mathbf{e}'$, plastic strains ${}^t\mathbf{e}^P$ and creep strains ${}^t\mathbf{e}^C$ are known from the current displacements and the stress/strain state at the start of the current time step.

The following scalar function $f({}^{t+\Delta t}\bar{\sigma})$ is obtained from Eq. (3.6-3)

$$f({}^{t+\Delta t}\bar{\sigma}) = a^2 {}^{t+\Delta t}\bar{\sigma}^2 + b {}^\tau\gamma - c^2 {}^\tau\gamma^2 - d^2 = 0 \quad (3.6-4)$$

The zero of equation (3.6-4) provides the solution for the effective stress ${}^{t+\Delta t}\bar{\sigma}$, where

$$a = {}^{t+\Delta t}a_E + \alpha \Delta t {}^\tau\gamma + \Delta\lambda$$

$$b = 3(1 - \alpha)\Delta t {}^{t+\Delta t}e'_{ij} {}^t s_{ij}$$

$$c = (1 - \alpha)\Delta t {}^t\bar{\sigma}$$

$$d^2 = \frac{3}{2} {}^{t+\Delta t}e''_{ij} {}^{t+\Delta t}e''_{ij}$$

with summation on the indices i, j .

Once the solution for ${}^{t+\Delta t}\bar{\sigma}$ has been determined from Eq. (3.6-4), simultaneously with the scalars ${}^\tau\gamma$ and $\Delta\lambda$ from the creep and plasticity conditions, the deviatoric stress ${}^{t+\Delta t}\mathbf{s}$ is calculated from Eq. (3.6-3), and the plastic and creep strains at the end of the time step are obtained as

$${}^{t+\Delta t}\mathbf{e}^P = {}^t\mathbf{e}^P + \Delta\lambda {}^{t+\Delta t}\mathbf{s}$$

$${}^{t+\Delta t}\mathbf{e}^C = {}^t\mathbf{e}^C + \left[(1-\alpha) {}^t\mathbf{s} + \alpha {}^{t+\Delta t}\mathbf{s} \right] \Delta t {}^\tau\gamma$$

The above equations correspond to isotropic hardening conditions and a general 3-D analysis. The solution details for kinematic hardening conditions and for special problems (for the plane stress, shell, isoparametric beam elements) are given in the above cited reference and also in the following reference.

ref. M. Kojić and K.J. Bathe, "Thermo-Elastic-Plastic and Creep Analysis of Shell Structures", *Computers & Structures*, Vol. 26, No 1/2, pp. 135-143, 1987.

3.6.5 Shifting rule for the irradiation creep model

- Irradiation tests obtaining irradiation strain data and material property changes of graphite have been carried out generally at a constant temperature. They cannot be determined directly from the test data due to variation of temperature.

Therefore, the irradiation strain at temperature θ and under fast neutron fluence ϕ is calculated as follows

$$\varepsilon^W(\theta, \phi) = \sum_{i=1}^n \Delta\varepsilon^W(\theta_i, \phi_{i-1}, \phi_i - \phi_{i-1})$$

where

$$\Delta\varepsilon^W(\theta_i, \phi, \Delta\phi) = \varepsilon^W(\theta_i, \phi + \Delta\phi) - \varepsilon^W(\theta_i, \phi)$$

This is illustrated in Figure 3.6-6, and documented in the following reference.

ref. T. Iyoku and M. Ishihara, "Development of Thermal/Irradiation Stress Analytical Code VIENUS for HTTR Graphite Block", *Journal of Nuclear Science and Technology*, 28[10], pp. 921-931, October 1991.

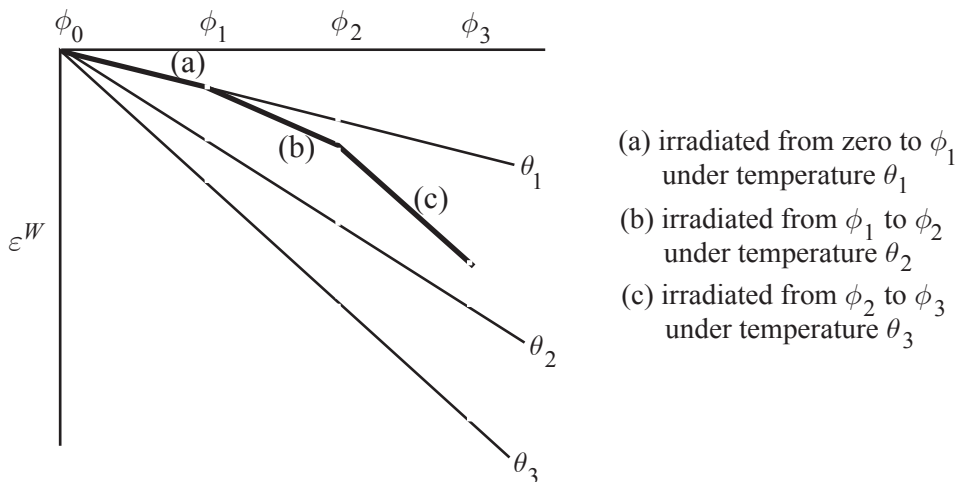


Figure 3.6-6: Shifting rule for irradiation creep model

3.7 Concrete material models

- The concrete material models in ADINA are the **concrete** and **data fitted concrete** material models.
- The concrete material models can be employed with the **2-D solid** and **3-D solid** elements, and **multi-layer (composite) shell** elements. The data fitted (DF) concrete material model can also be employed with **3-D shell** elements.
- The concrete model can be used with the **small displacement** and **large displacement** formulations. When used with the small displacement formulation, a materially-nonlinear-only (MNO) formulation is employed, and when used with the large displacement formulation, a total Lagrangian (TL) formulation is employed.
- These material models are supported in explicit and implicit analyses.
- These material models allow three types of temperature effects:
 - ▶ no temperature effects;
 - ▶ isotropic temperature independent effects; and

- ▶ isotropic temperature dependent effects.
- Reinforcement bars can be included in the concrete model using rebar elements. Rebar elements are truss elements that are automatically connected to any 2-D or 3-D solid mesh in which truss elements lie. The AUI does this connection using constraint equations during data file generation. For more information on rebar elements, see Section 2.1.7.
- Rebar elements of Section 2.1.7 can be incorporated into 2-D or 3-D solid elements for modeling as concrete structures. The reinforcements in the concrete shell can employ various material models, such as the elastic-orthotropic, plastic-orthotropic models, into the shell layers.

3.7.1 Concrete material model

- Although the model is entitled "concrete model", the basic constitutive characteristics are such that the model can also be useful when representing other materials. The basic material characteristics are:
 - ▶ Tensile cracking failure at a maximum, relatively small principal tensile stress
 - ▶ Compression crushing failure at high compression
 - ▶ Strain softening from compression crushing failure to an ultimate strain, at which the material totally fails.

The tensile cracking and compression crushing failures are governed by tensile failure and compression crushing failure envelopes.

These material characteristics pertain, for example, to a variety of rocks.

- It is well accepted that concrete is a very complex material. The model provided in ADINA may not contain all the detailed material characteristics that you may look for. However, considering the variability of concrete materials that need to be described in practice, and recognizing that the model may also be useful in the

modeling of rock materials, the objective is to provide an effective model with sufficient flexibility to model most of the commonly used material behaviors.

- The notation used in the formulation of the model is given below.

tE = equivalent multiaxial tangent Young's modulus at time t (the left superscript " t " refers to time t)

\tilde{E}_0 = uniaxial initial tangent modulus (all uniaxial quantities are identified with a tilde " \sim ")

\tilde{E}_s = uniaxial secant modulus corresponding to uniaxial maximum stress, $\tilde{E}_s = \frac{\tilde{\sigma}_c}{\tilde{\epsilon}_c}$

\tilde{E}_u = uniaxial secant modulus corresponding to uniaxial ultimate stress, $\tilde{E}_u = \frac{\tilde{\sigma}_u}{\tilde{\epsilon}_u}$

${}^t\tilde{E}_{pi}$ = uniaxial tangent modulus in the direction of ${}^t\sigma_{pi}$

${}^te_{ij}$ = total strains

e_{ij} = incremental strains

${}^t\tilde{\epsilon}$ = uniaxial strain

$\tilde{\epsilon}_c$ = uniaxial strain corresponding to $\tilde{\sigma}_c$ ($\tilde{\epsilon}_c < 0$)

$\tilde{\epsilon}_u$ = ultimate uniaxial compressive strain ($\tilde{\epsilon}_u < 0$)

${}^t\sigma_{ij}$ = total stresses

σ_{ij} = incremental stresses

${}^t\tilde{\sigma}$ = uniaxial stress

$\tilde{\sigma}_t$ = uniaxial cut-off tensile stress ($\tilde{\sigma}_t > 0$)

$\tilde{\sigma}_{tp}$ = post-cracking uniaxial cut-off tensile stress ($0 \leq \tilde{\sigma}_{tp} \leq \tilde{\sigma}_t$). Note that if $\tilde{\sigma}_{tp} = 0$, ADINA sets $\tilde{\sigma}_{tp} = \tilde{\sigma}_t$.

$\tilde{\sigma}_c$ = maximum uniaxial compressive stress ($\tilde{\sigma}_c < 0$)

$\tilde{\sigma}_u$ = ultimate uniaxial compressive stress ($\tilde{\sigma}_u < 0$)

${}^t\sigma_{pi}$ = principal stress in direction i (${}^t\sigma_{p1} \geq {}^t\sigma_{p2} \geq {}^t\sigma_{p3}$)

- Three basic features are used in the concrete model:
 - ▶ A nonlinear stress-strain relation to allow for the weakening of the material under increasing compressive stresses
 - ▶ Failure envelopes that define cracking failure in tension and crushing in compression
 - ▶ A strategy to model the post-cracking and crushing behavior of the material

Stress-strain relations: The general multiaxial stress-strain relations are derived from a uniaxial stress-strain relation ${}^t\tilde{\sigma}$ versus ${}^t\tilde{\epsilon}$.

A typical uniaxial stress ${}^t\tilde{\sigma}$ to uniaxial strain ${}^t\tilde{\epsilon}$ relation is shown in Fig. 3.7-1. This stress-strain relation shows that there are three strain phases; namely, ${}^t\tilde{\epsilon} \geq 0$, $0 > {}^t\tilde{\epsilon} \geq \tilde{\epsilon}_c$, and $\tilde{\epsilon}_c > {}^t\tilde{\epsilon} \geq \tilde{\epsilon}_u$ where $\tilde{\epsilon}_c$ is the strain corresponding to the minimum (crushing) stress $\tilde{\sigma}_c$ that can be reached, and $\tilde{\epsilon}_u$ is the ultimate compressive strain.

If ${}^t\tilde{\epsilon} > 0$, the material is in tension and the stress-strain relation is linear until tensile failure at the stress $\tilde{\sigma}_t$. A constant Young's modulus \tilde{E}_0 is employed, i.e.,

$${}^t\tilde{\sigma} = \tilde{E}_0 {}^t\tilde{\epsilon} \quad (3.7-1)$$

For ${}^t\tilde{\epsilon} \leq 0$, we assume the following relation

$$\frac{{}^t\tilde{\sigma}}{\tilde{\sigma}_c} = \frac{\left(\frac{\tilde{E}_0}{\tilde{E}_s}\right)\left(\frac{{}^t\tilde{e}}{\tilde{e}_c}\right)}{1 + A\left(\frac{{}^t\tilde{e}}{\tilde{e}_c}\right) + B\left(\frac{{}^t\tilde{e}}{\tilde{e}_c}\right)^2 + C\left(\frac{{}^t\tilde{e}}{\tilde{e}_c}\right)^3}$$

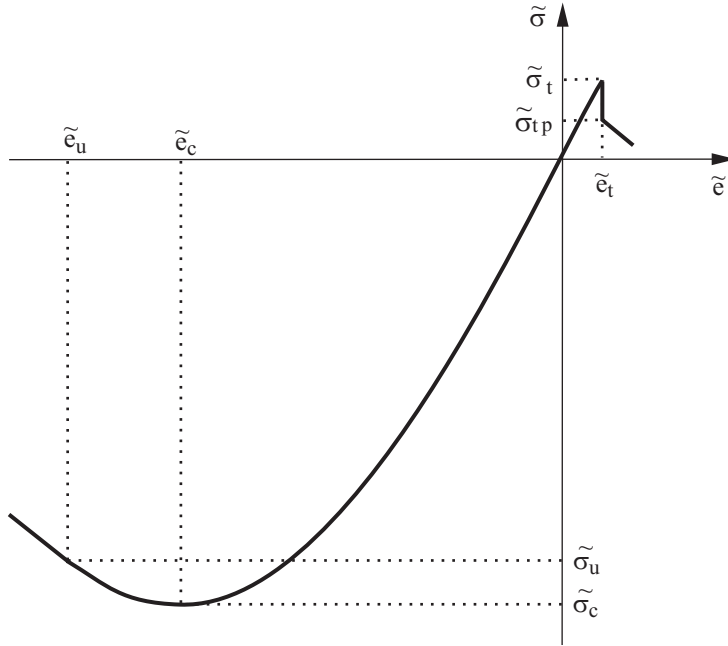


Figure 3.7-1: Uniaxial stress-strain relation used in the concrete model

and hence

$${}^t\tilde{E} = \frac{\tilde{E}_0 \left[1 - B\left(\frac{{}^t\tilde{e}}{\tilde{e}_c}\right)^2 - 2C\left(\frac{{}^t\tilde{e}}{\tilde{e}_c}\right)^3 \right]}{\left[1 + A\left(\frac{{}^t\tilde{e}}{\tilde{e}_c}\right) + B\left(\frac{{}^t\tilde{e}}{\tilde{e}_c}\right)^2 + C\left(\frac{{}^t\tilde{e}}{\tilde{e}_c}\right)^3 \right]^2} \quad (3.7-2)$$

where

$$A = \frac{\left[\frac{\tilde{E}_0}{\tilde{E}_u} + (p^3 - 2p^2) \frac{\tilde{E}_0}{\tilde{E}_s} - (2p^3 - 3p^2 + 1) \right]}{\left[(p^2 - 2p + 1)p \right]}$$

$$B = \left[\left(2 \frac{\tilde{E}_0}{\tilde{E}_s} - 3 \right) - 2A \right]$$

$$C = \left[\left(2 - \frac{\tilde{E}_0}{\tilde{E}_s} \right) + A \right]$$

and the parameters \tilde{E}_0 , $\tilde{\sigma}_c$, \tilde{e}_c , $\tilde{E}_s = \frac{\tilde{\sigma}_c}{\tilde{e}_c}$, $\tilde{\sigma}_u$, \tilde{e}_u , $p = \frac{\tilde{e}_u}{\tilde{e}_c}$, $\tilde{E}_u = \frac{\tilde{\sigma}_u}{\tilde{e}_u}$

are obtained from uniaxial tests.

The stress-strain relation in Eq. (3.7-2) assumes monotonic loading conditions. For unloading conditions and loading back to the stress state from which unloading occurred, the initial Young's modulus \tilde{E}_0 is used. For strain states beyond \tilde{e}_u in compression, it is assumed that stresses are linearly released to zero, using the following modulus:

$$\tilde{E}_u = \frac{\tilde{\sigma}_u - \tilde{\sigma}_c}{\tilde{e}_u - \tilde{e}_c}$$

Note that confined concrete can therefore be modeled using close values for $\tilde{\sigma}_u$ and $\tilde{\sigma}_c$.

Under multiaxial stress conditions, the stress-strain relations are evaluated differently depending on whether the material is loading or unloading. Poisson's ratio is assumed to be constant under tensile stress conditions and can vary in the compressive region.

To characterize loading and unloading conditions we define a loading scalar ${}^t g$ for each integration point,

$${}^t g = {}^t \sigma_e \quad (3.7-3)$$

where ${}^t \sigma_e$ is the effective stress at time t . The material is loading at the integration point except when the unloading conditions are determined,

$${}^t g < g_{max} \quad (3.7-4)$$

where g_{max} is the maximum value of the loading scalar that has been reached during the complete solution.

During unloading, the material is assumed to be isotropic and the initial Young's modulus, \tilde{E}_0 , is used to form the incremental stress-strain matrix, both for stiffness and stress calculations.

In loading conditions, the principal stresses ${}^t \sigma_{pi}$ (with ${}^t \sigma_{p1} \geq {}^t \sigma_{p2} \geq {}^t \sigma_{p3}$) are calculated. For each principal stress direction, a tangent Young's modulus ${}^t \tilde{E}_{pi}$ corresponding to the given strain state ${}^t e_{pi}$ is evaluated using Eqs. (3.7-1) and (3.7-2).

In Eq. (3.7-2), the strains ${}^t e_{pi}$ are used together with the parameters $\tilde{\sigma}'_c$, $\tilde{\sigma}'_u$, $\tilde{\epsilon}'_c$ and $\tilde{\epsilon}'_u$ defined in Eq. (3.7-8) which accounts for the multiaxial stress conditions.

The material is considered as orthotropic with the directions of orthotropy being defined by the principal stress directions. Once cracking occurs in any direction i , that direction is fixed from that point onward in calculating ${}^t \sigma_{pi}$.

The stress-strain matrix corresponding to these directions is, for three-dimensional stress conditions,

$$\mathbf{C} = \frac{1}{(1+\nu)(1-2\nu)} \times \begin{bmatrix} (1-\nu) {}^t \tilde{E}_{p1} & \nu {}^t E_{12} & \nu {}^t E_{13} & 0 & 0 & 0 \\ & (1-\nu) {}^t \tilde{E}_{p2} & \nu {}^t E_{23} & 0 & 0 & 0 \\ & & (1-\nu) {}^t \tilde{E}_{p3} & 0 & 0 & 0 \\ & & & 0.5(1-2\nu) {}^t E_{12} & 0 & 0 \\ \text{symmetric} & & & & 0.5(1-2\nu) {}^t E_{13} & 0 \\ & & & & & 0.5(1-2\nu) {}^t E_{23} \end{bmatrix} \quad (3.7-5)$$

where ν is the constant Poisson's ratio and the ${}^t E_{ij}$ with $i \neq j$ are

evaluated using

$${}^tE_{ij} = \frac{|{}^t\sigma_{pi}| {}^t\tilde{E}_{pi} + |{}^t\sigma_{pj}| {}^t\tilde{E}_{pj}}{|{}^t\sigma_{pi}| + |{}^t\sigma_{pj}|} \quad (3.7-6)$$

Note that the above stress-strain relations for material loading conditions are only employed in the calculation of the stiffness matrix at time t . Considering the evaluation of the stress increment from t to time $t + \Delta t$, we have

$$\boldsymbol{\sigma} = \hat{\mathbf{C}} \mathbf{e} \quad (3.7-7)$$

where the stress-strain matrix $\hat{\mathbf{C}}$ is the one defined in Eq. (3.7-5) but using the Young's moduli ${}^t\tilde{E}_{pi}$. These moduli are evaluated using three point Gaussian integration of \tilde{E}_{pi} between ${}^te_{pi}$ and ${}^{t+\Delta t}e_{pi}$, with ${}^te_{pi}$ and ${}^{t+\Delta t}e_{pi}$ the strain components measured in the directions of the respective principal stresses ${}^t\sigma_{pi}$ and ${}^{t+\Delta t}\sigma_{pi}$.

Material failure envelopes: The failure envelopes shown in Figs. 3.7-2 to 3.7-5 are employed to establish the uniaxial stress-strain law accounting for multiaxial stress conditions, and to identify whether tensile or crushing failure of the material has occurred.

Uniaxial stress-strain law under multiaxial stress conditions:

Having established the principal stresses ${}^t\sigma_{pi}$ with

${}^t\sigma_{p1} \geq {}^t\sigma_{p2} \geq {}^t\sigma_{p3}$ the stresses ${}^t\sigma_{p1}$ and ${}^t\sigma_{p2}$ are held constant and the minimum stress that would have to be reached in the third principal direction to cause crushing of the material is calculated using the failure envelope, see Fig. 3.7-5(a). Let this stress be $\tilde{\sigma}'_c$, we have

$$\gamma_1 = \frac{\tilde{\sigma}'_c}{\tilde{\sigma}_c}$$

and

$$\begin{aligned}\tilde{\sigma}'_u &= \gamma_1 \tilde{\sigma}_u; & \tilde{e}'_c &= (C_1 \gamma_1^2 + C_2 \gamma_1) \tilde{e}_c; \\ \tilde{e}'_u &= (C_1 \gamma_1^2 + C_2 \gamma_1) \tilde{e}_u\end{aligned}\quad (3.7-8)$$

where C_1 and C_2 are input parameters. Normally, $C_1 = 1.4$ and $C_2 = -0.4$. The constants $\tilde{\sigma}'_c$, $\tilde{\sigma}'_u$, \tilde{e}'_c and \tilde{e}'_u are employed instead of the unprimed variables in order to establish, using Eq. (3.7-2), the uniaxial stress-strain law under multiaxial conditions (see Fig. 3.7-5(b)).

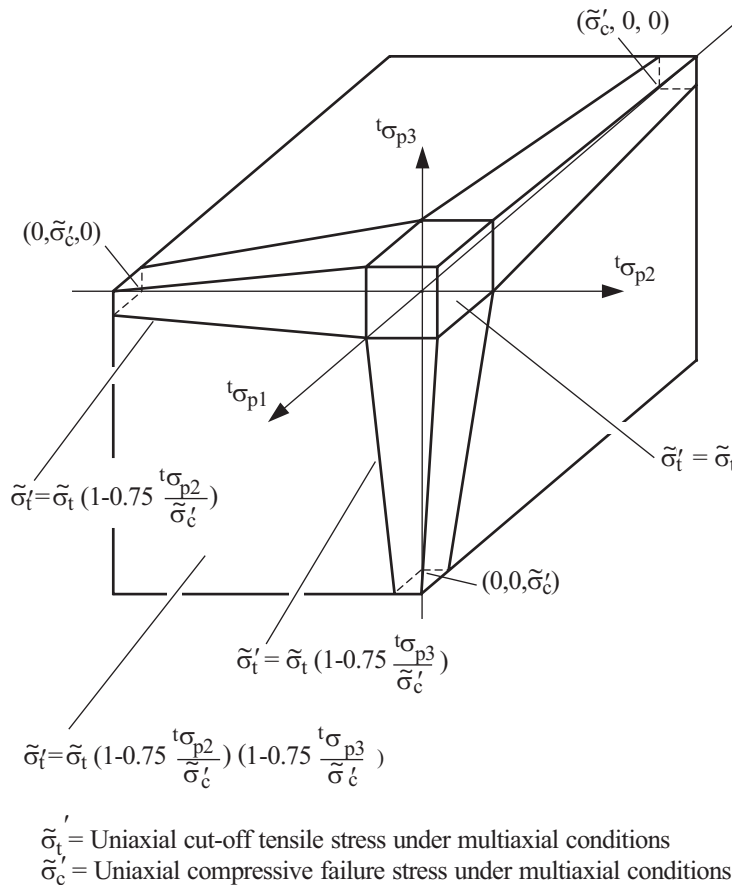


Figure 3.7-2: Three-dimensional tensile failure envelope of concrete model

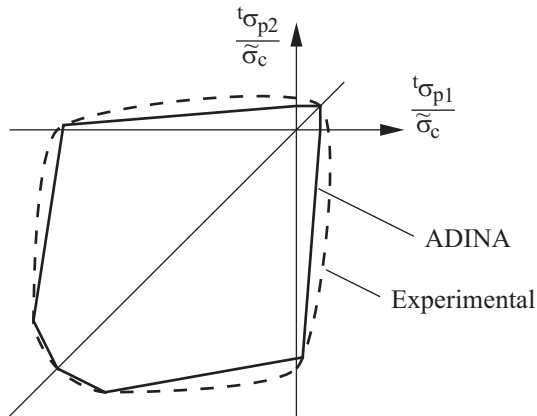


Figure 3.7-3: Biaxial concrete compressive failure envelope

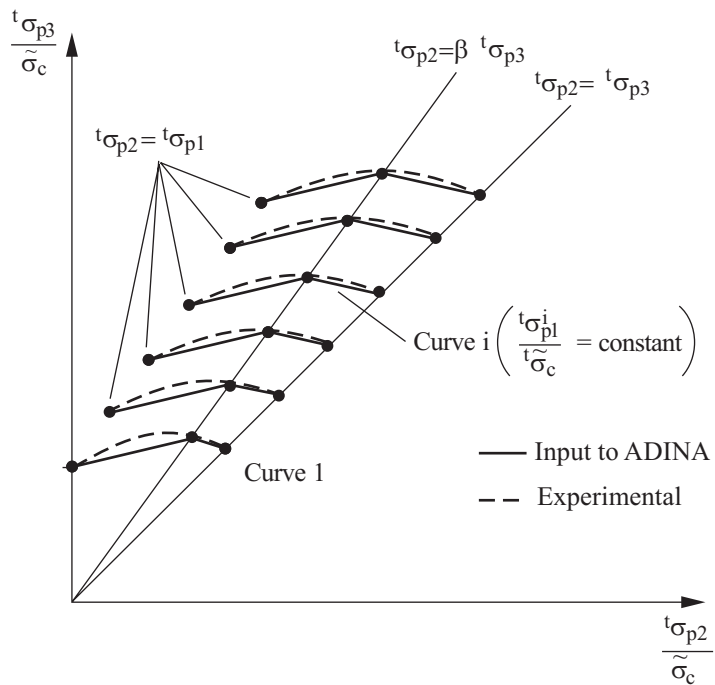
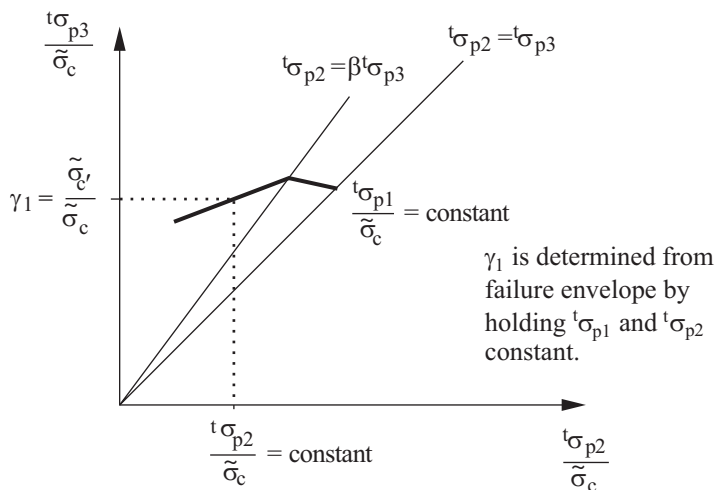
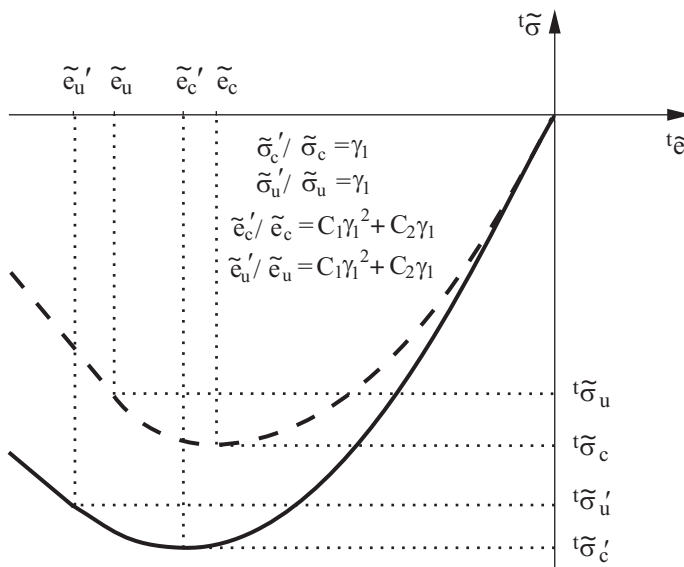


Figure 3.7-4: Triaxial compressive failure envelope



(a) Determination of $\tilde{\sigma}_c'$ from given $({}^t\sigma_{p1}, {}^t\sigma_{p2})$



(b) Stress-strain behavior under multiaxial stress conditions

Figure 3.7-5: Definitions used for evaluation of one-dimensional stress-strain law under multiaxial stress conditions

Tensile failure envelope: The tensile failure envelope used in the concrete model is shown in Fig. 3.7-2. To identify whether the material has failed, the principal stresses are used to locate the current stress state. Note that the tensile strength of the material in a principal direction does not depend on tensile stresses in the other principal stress directions, but depends on compressive stresses in the other directions.

Tensile failure occurs if the tensile stress in a principal stress direction exceeds the tensile failure stress. In this case it is assumed that a plane of failure develops perpendicular to the corresponding principal stress direction. The effect of this material failure is that the normal and shear stiffnesses and stresses across the plane of failure are reduced and plane stress conditions are assumed to exist at the plane of tensile failure, as discussed in more detail below.

Prior to tensile failure the stress-strain material law is given by Eqs. (3.7-5) to (3.7-7). Assuming that ${}^t\sigma_{pi}$ is larger than the tensile failure stress, the new material stress-strain matrix used in the stiffness matrix calculation is given by

$$\mathbf{C} = \begin{bmatrix} \tilde{E}_0\eta_n & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{(1-\nu^2)} \begin{bmatrix} {}^t\tilde{E}_{p2} & \nu {}^tE_{23} \\ {}^t\tilde{E}_{p3} & {}^t\tilde{E}_{p3} \end{bmatrix} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \frac{\tilde{E}_0\eta_s}{2(1+\nu)} & 0 & 0 \\ \text{symmetric} & \frac{\tilde{E}_0\eta_s}{2(1+\nu)} & 0 \\ \frac{{}^tE_{23}}{2(1+\nu)} \end{bmatrix} \quad (3.7-9)$$

where the ${}^t\tilde{E}_{pi}$ are the uniaxial Young's moduli evaluated in the principal stress directions using Eqs. (3.7-1) or (3.7-2), and the ${}^tE_{ij}$ are evaluated using Eq. (3.7-6).

The constants η_n and η_s are the normal and shear stiffness reduction factors, respectively. Typically, $\eta_n = 0.0001$ and $\eta_s = 0.5$. The factor η_n is not exactly equal to zero in order to avoid the possibility of a singular stiffness matrix. The factor η_s depends on a number of physical factors and you must use judgement to choose its value. For the concrete model in ADINA, η_n and η_s are both input parameters.

For the stress calculation, the following stress-strain matrices are used:

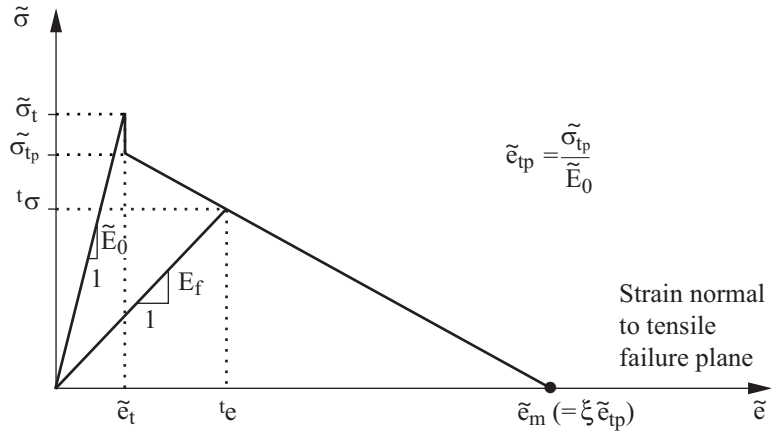
- For the tensile stress normal to the tensile failure plane and the shear stresses in this plane, we use the total strains to calculate the total stresses with

$$\hat{\mathbf{C}} = \begin{bmatrix} E_f & 0 & 0 \\ & G_{12}^f & 0 \\ \text{sym.} & & G_{13}^f \end{bmatrix} \quad (3.7-10a)$$

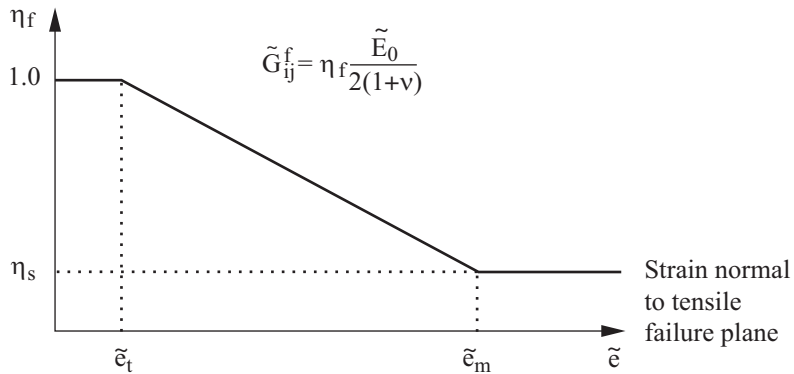
where E_f and G_{12}^f, G_{13}^f are evaluated as shown in Fig. 3.7-6. In this figure, ξ is a user input variable that defines the amount of tension stiffening. Fig. 3.7-6(b) shows that G_{ij}^f is evaluated from the initial shear modulus. Also, Fig. 3.7-6(a) assumes loading from zero stress directly into the tensile region. If the tensile stress is reached by unloading from a compressive stress, the strain normal to the tensile failure plane is measured from the specific strain value at which the stress is zero (see Fig. 3.7-7 – point 11).

Special care in the analysis must be taken if ξ is chosen to be greater than 1.0, because strain softening may yield non-unique solutions.

To obtain a mesh independent solution, the fracture energy G_f can be provided instead of ξ . In doing so, ξ is evaluated at each integration point, based on the size of the finite elements (see Fig. 3.7-6(c)). In this case, ADINA also calculates η_s internally, overwriting the user input.

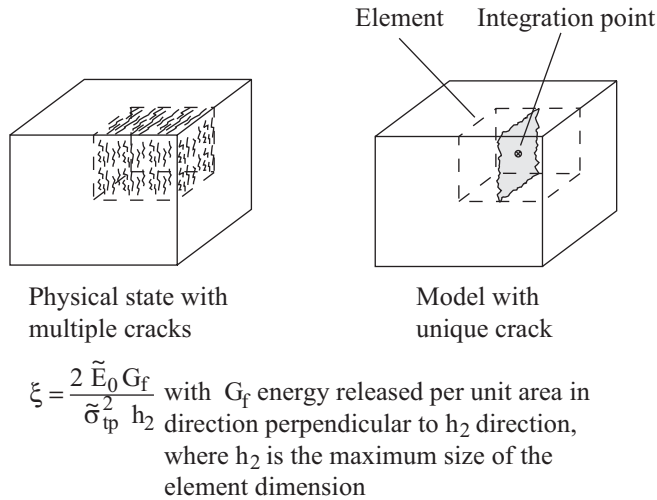


- a) Calculation of Young's modulus E_f , normal to tensile failure plane.
 $\tilde{\sigma}_{tp}$ is the post-cracking uniaxial cut-off tensile stress.



- b) Calculation of shear modulus in tensile failure plane

Figure 3.7-6: Material moduli for stress calculation after tensile failure



c) Smeared crack approach for the calculation of the parameter ξ

Figure 3.7-6: (continued)

- For the remaining stress components, the increments are evaluated from the strain increments using

$$\mathbf{C} = \frac{1}{1 - \nu^2} \begin{bmatrix} {}^t\tilde{E}_{p2} & \nu {}^t\tilde{E}_{23} & 0 \\ & {}^t\tilde{E}_{p3} & 0 \\ \text{sym.} & & (1 - \nu^2) \frac{{}^t\tilde{E}_{23}}{2(1 + \nu)} \end{bmatrix} \quad (3.7-10b)$$

where the ${}^t\tilde{E}_{pi}$ are the uniaxial Young's moduli used in Eq. (3.7-7) and ${}^t\tilde{E}_{23}$ is evaluated using Eq. (3.7-6) but with ${}^t\tilde{E}_{pi}$ instead of ${}^t\tilde{E}_{pi}$.

Compressive failure envelope: The triaxial failure envelope used in the concrete model is shown in Fig. 3.7-4. Note that the biaxial failure envelope shown in Fig. 3.7-3 is curve 1 of the triaxial failure envelope. Also, note that the biaxial and triaxial envelope

curves can be used to represent a large number of specific envelopes by the input explained below, and this flexibility makes it possible to model various concrete and rock materials.

The triaxial compression failure envelope is shown in Fig.

3.7-4. First, the values $\frac{\sigma_{p1}^i}{\bar{\sigma}_c}$ are input. These values define for

which levels of the ${}^t\sigma_{p1}$ the two-dimensional failure envelopes functions of stresses ${}^t\sigma_{p2}$ and ${}^t\sigma_{p3}$ are input.

Six failure envelopes must be defined, each with three points corresponding to the locations ${}^t\sigma_{p2} = {}^t\sigma_{p1}$, ${}^t\sigma_{p2} = \beta {}^t\sigma_{p3}$, and ${}^t\sigma_{p2} = {}^t\sigma_{p3}$ where β is an input parameter ($0 \leq \beta \leq 1$).

To identify compression failure, the largest principal stress, ${}^t\sigma_{p1}$, is employed to establish the biaxial failure envelope function of ${}^t\sigma_{p2}$ and ${}^t\sigma_{p3}$, using the interpolation shown in Fig. 3.7-5(a). If the stress state corresponding to ${}^t\sigma_{p2}$ and ${}^t\sigma_{p3}$ lies on or outside this biaxial failure envelope, then the material has crushed.

It should be noted that you must choose appropriate values for the input of the failure surfaces and other parameters. These may vary significantly for different materials and structures and must be obtained from experimental data.

Material behavior after failure: The post failure material behaviors considered in the concrete model in ADINA include the post tensile cracking, post compression crushing, and strain-softening behaviors.

Post tensile cracking behavior: Once a tensile plane of failure has formed, it is checked in each subsequent solution step to see whether the failure is still active. The failure is considered to be inactive provided the normal strain across the plane becomes negative and less than the strain at which the "last" failure occurred. It is otherwise active. Therefore, a tensile failure plane can repeatedly be active and inactive. Consider the uniaxial cyclic loading case shown in Fig. 3.7-7 in which the strain is prescribed. As the strain reaches the tensile cut-off limit, a tensile failure plane becomes active. This tensile failure plane remains active while the

strain continues to increase to point 2 and then decreases to point 3 and then increases to point 4 and then decreases to point 5. The failure plane becomes inactive after 5 and remains so while the strain decreases to point 8 and increases again to point 11. At point 11, the tensile failure plane again becomes active, and it remains active until the strain reaches point 13 and beyond.

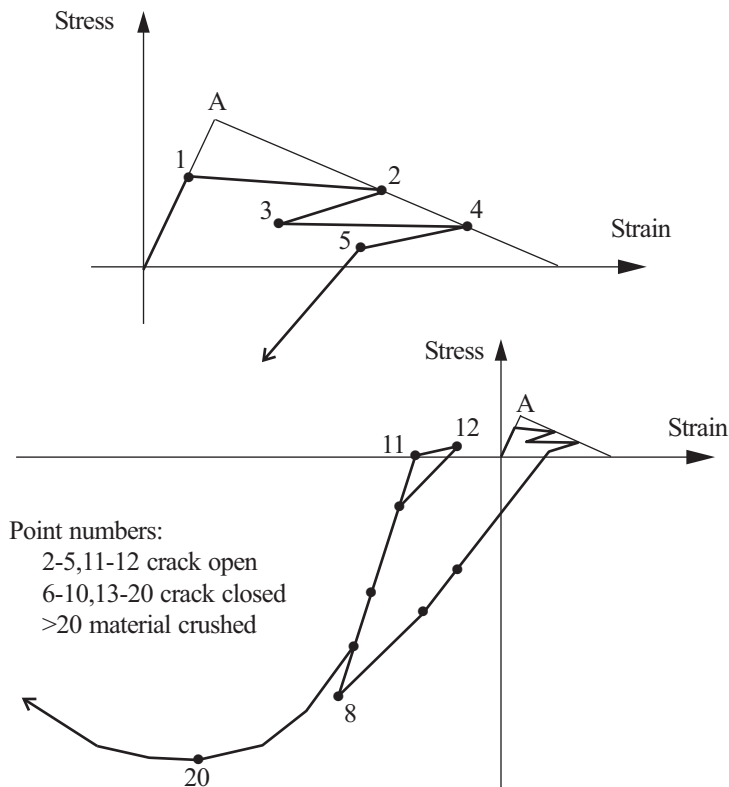


Figure 3.7-7: Uniaxial cyclic loading stress/strain experiment with the concrete model

If a tensile failure plane has developed, which may or may not be active, the material stress-strain relations are always established as described above, but corresponding to the directions along and perpendicular to the plane of failure. Hence, instead of using the principal stresses and corresponding directions as done for the unfailed material, the stresses in the directions defined by the

material tensile failure plane are used to evaluate the stress-strain matrix, corresponding to those directions. Once a failure plane has been initiated, a subsequent failure plane is assumed to form perpendicular to the direction of the first failure plane whenever a normal stress along the original failure plane has reached the tensile failure stress. It follows that at any integration point, the direction of the third tensile failure plane is fixed once failure has occurred in two directions.

Post compression crushing, strain-softening behavior: If the material has crushed in compression, it is assumed that the material strain-softens in all directions.

Consider first uniaxial stress conditions. As shown in Fig. 3.7-1, when the uniaxial strain is smaller than $\tilde{\epsilon}_c$, the material has crushed and softens with increasing compressive strain, i.e., ${}^t\tilde{E}$ is negative.

Under multiaxial stress conditions the compression crushing is identified using the multiaxial failure envelope, and once the material has crushed, isotropic conditions are assumed. As in uniaxial conditions, in the subsequent solution steps the Young's modulus is assumed to be very small but positive in the stiffness matrix calculations, but the stress increments are computed from the uniaxial stress-strain law with the constants $\tilde{\sigma}'_c$, $\tilde{\epsilon}'_c$ and so on, see Eq. (3.7-8), corresponding to the multiaxial conditions at crushing. The Young's modulus tE corresponding to the current strain increment e_{p3} is evaluated using the uniaxial stress-strain relationship in Fig. 3.7-1

$${}^tE = \frac{\left[\tilde{\sigma}'_c \big|_{{}^te_{p3} + e_{p3}} - \tilde{\sigma}'_c \big|_{{}^te_{p3}} \right]}{e_{p3}}$$

where ${}^te_{p3}$ and e_{p3} are the strain component and incremental strain components at time t measured in the direction of the principal stress ${}^t\sigma_{p3}$. To obtain the stress increment, Eq. (3.7-7) is used where the matrix $\hat{\mathbf{C}}$ corresponds to isotropic material conditions with Young's modulus tE . Note that when ${}^te_{p3}$ becomes equal to or less than $\tilde{\epsilon}'_u$, the stresses are linearly released to zero.

If unloading of the crushed material in the strain-softening region occurs, characterized by $e_{p3} \geq 0$, the initial Young's modulus \tilde{E}_0 is used. At any time during post-crushing calculations, if any one of the principal stresses checked individually reaches a positive value, this stress is set to zero.

Temperature effects: In some analyses, thermal strains ${}^t\mathbf{e}^{TH}$ need to be included. These strains are taken into account by replacing the total incremental strain \mathbf{e} in the governing incremental equations with the strain $\mathbf{e} - \mathbf{e}^{TH}$, where \mathbf{e}^{TH} is the thermal incremental strain and is calculated from the temperature conditions.

The following material properties can be defined as temperature-dependent:

- ▶ Uniaxial initial tangent modulus \tilde{E}_0
- ▶ Poisson's ratio ν
- ▶ Mean coefficient of thermal expansion α
- ▶ Uniaxial cut-off tensile stress $\tilde{\sigma}_t$
- ▶ Post-cracking uniaxial cut-off tensile stress $\tilde{\sigma}_{tp}$
- ▶ Uniaxial maximum compressive stress $\tilde{\sigma}_c$
- ▶ Uniaxial compressive strain at $\tilde{\sigma}_c$, $\tilde{\epsilon}_c$
- ▶ Uniaxial ultimate compressive stress $\tilde{\sigma}_u$
- ▶ Uniaxial strain at $\tilde{\sigma}_u$, $\tilde{\epsilon}_u$
- ▶ Fracture energy G_f
- ▶ Constant for tensile strain failure ξ

The nodal point temperatures are input to ADINA as discussed in Section 5.6.

Poisson's ratio in the compressive region: It has been observed in experiments that the ratio of lateral strain to principal compressive strain remains constant until approximately 80% of the maximum compressive stress $\tilde{\sigma}_c$.

Usually, we assume that the Poisson ratio is constant throughout the analysis; however, as an option, the value ν_s can be used when

the material dilates under compression. The value of ν_s is given by

$$\begin{cases} \nu_s = \nu & \text{when } \gamma_2 = \frac{{}^t\sigma_{p3}}{\tilde{\sigma}'_c} \leq \gamma_a \\ \nu_s = \nu_f - (\nu_f - \nu) \sqrt{1 - \left(\frac{\gamma_2 - \gamma_a}{1 - \gamma_a} \right)^2} & \text{when } \gamma_2 > \gamma_a \end{cases}$$

where ν is the initial Poisson's ratio, $\nu_f = 0.42$ is the maximum Poisson's ratio at failure and $\gamma_a = 0.7$.

3.7.2 Data fitted (DF) concrete material model

- The data fitted concrete material model is, in essence, an empirical material model that identifies concrete as a brittle material, which behaves nonlinearly in compression, and that it can be solely described by its uniaxial cylinder compressive strength (i.e., the maximum strength that a cylinder specimen of concrete can sustain when it is subjected to uniaxial compressive loading).
- The model is based on the work of M.D. Kotsovos and other researchers, and is described in the following references:
 - ref. M. D. Kotsovos and M. N. Pavlović, *Structural Concrete: Finite-element analysis for limit-state design*, Thomas Telford, London, 1995.
 - ref. M. D. Kotsovos and K. V. Spiliopoulos, “Modeling of crack closure for finite-element analysis of structural concrete”, *Computers & Structures*, 69:383–398, 1998.
- The experimental data reported by Kotsovos et al. indicate that one can attribute the following material characteristics to plain concrete:
 - can be considered an isotropic material with a nonlinear behavior in compression followed by a brittle post-peak behavior, which is characterized by a complete and immediate

loss of load-carrying capacity after the ultimate strength is attained;

- ▶ cannot be considered to be a continuum medium beyond its peak load level since Poisson's ratio values greater than 0.5 were measured near and around the peak load level;
 - ▶ microcracking fracture processes define the compressive behavior of concrete up to failure (in which the microcracks extend in the direction of the maximum compressive principal stress), while macrocracking describes the fracture processes occurring during failure (in which crack planes form such that they are orthogonal to the maximum tensile principal stress existing just prior to cracking);
 - ▶ no appreciable stress-strain hysteresis loop occurs during unloading and reloading;
 - ▶ failure can be captured in the stress space via a failure surface and it usually occurs first in all possible triaxial states of stress except the fully compressive triaxial stress state;
 - ▶ proper characterization of concrete behavior requires the description of general triaxial stress states, which are conveniently interpreted by the octahedral stresses;
 - ▶ stress path independence characterizes the experimental data that define the stress-strain curves as well as the failure surface;
 - ▶ all material constants can be related to the uniaxial compressive strength of a cylinder specimen; and
 - ▶ the overall mechanical properties are independent of the loading rate.
- In consideration of the previous premise, the data fitted concrete material model consists of
 - (i) a nonlinear experiment based stress-strain law in compression that is combined with a linear behavior when concrete unloads/reloads and is in tension (see Fig 3.7-8);

(ii) an experiment based stress failure surface function that dictates when either cracking or crushing occurs in concrete (see Fig. 3.7-9); and

(iii) a post-failure response for cracking based on a smeared crack approach that allows cracks to close and reopen (see Figures 3.7-10 to 3.7-12).

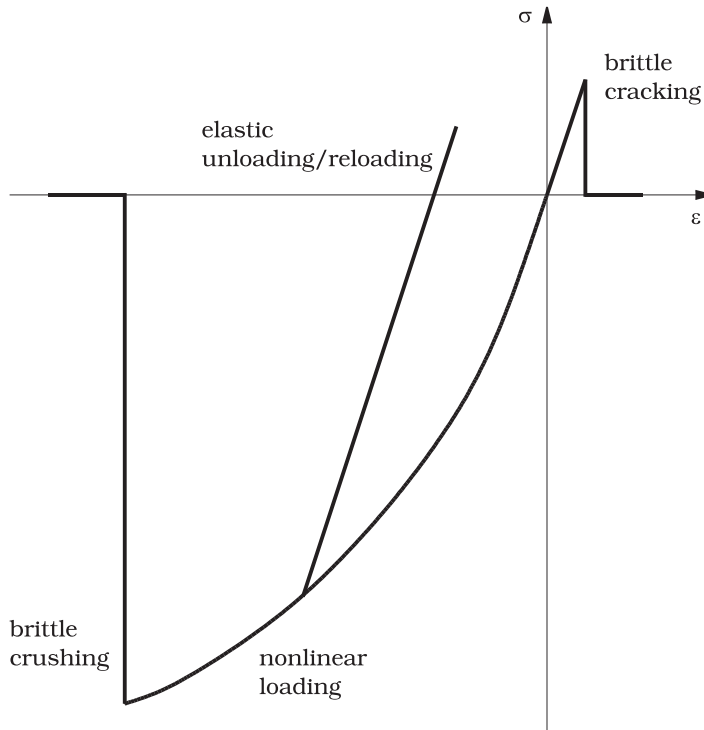


Figure 3.7-8: Stress-strain behavior of the data fitted concrete material model

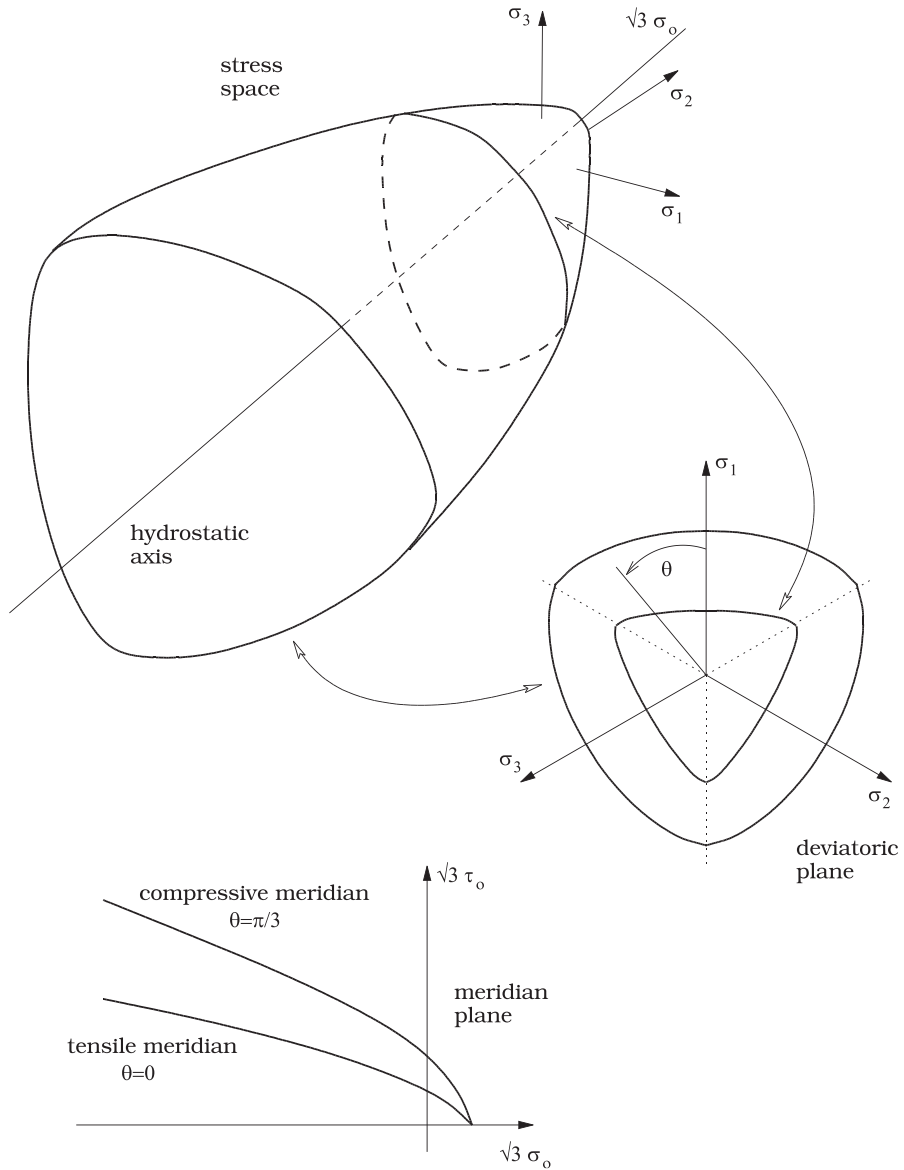


Figure 3.7-9: Failure surface of the data fitted concrete material model

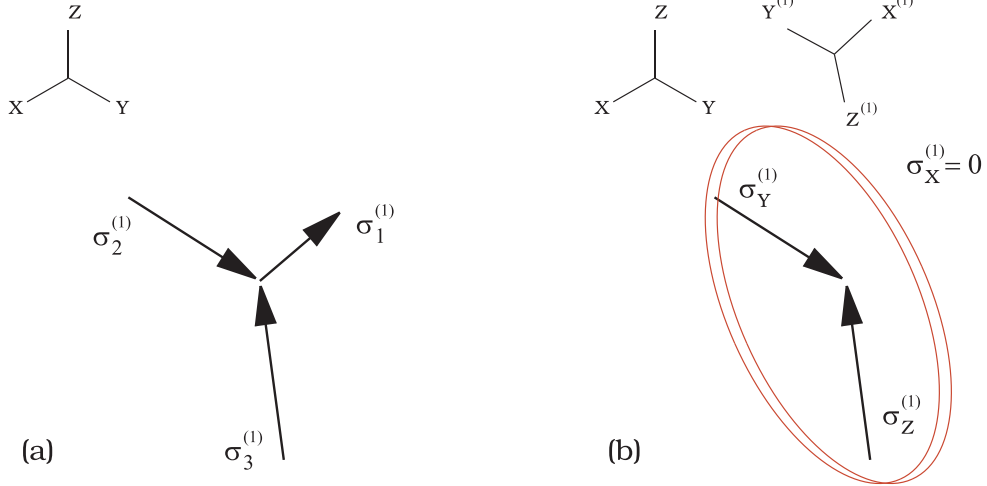


Figure 3.7-10: First crack formation of the data fitted concrete material model (a) before, and (b) after

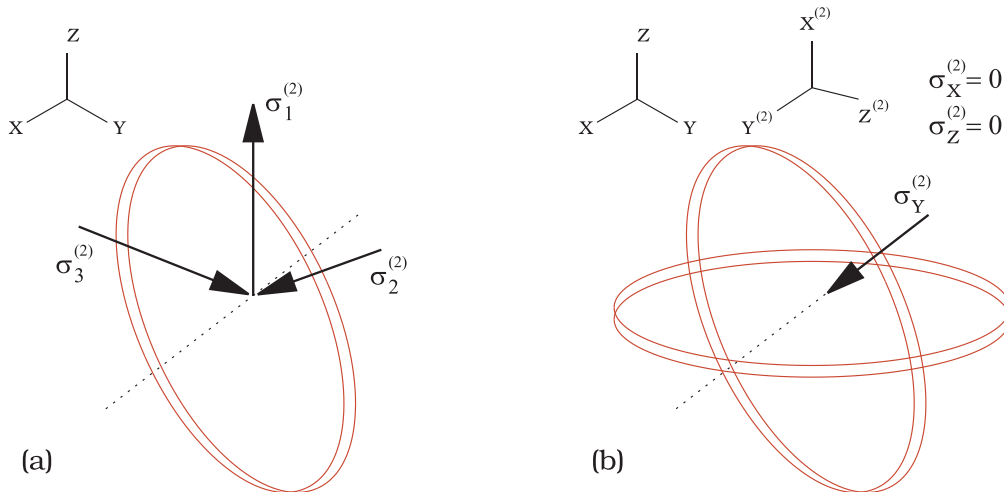


Figure 3.7-11: Second crack formation of the data fitted concrete material model (a) before, and (b) after

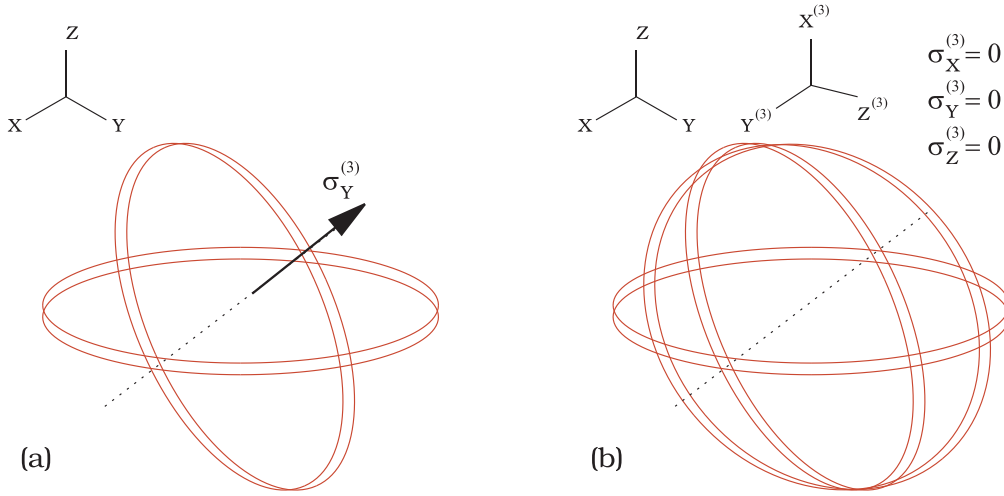


Figure 3.7-12: Third crack formation of the data fitted concrete material model (a) before, and (b) after

- The experimental data reported by Kotsovos et al. is presented in a compact way in terms of the octahedral normal and shear stresses (σ_o, τ_o) , their corresponding octahedral strains $(\varepsilon_o, \gamma_o)$, and the uniaxial cylinder compressive strength (f_c) material constant. Note that the octahedral normal and shear stresses possess a geometric interpretation in stress space similar to that of the first stress invariant and second deviatoric stress invariant (I_1, J_2) , respectively.
- Recall that any stress state is a point in the stress space that can be represented by the principal stresses $(\sigma_1, \sigma_2, \sigma_3)$ or, equivalently, by the stress invariants (I_1, J_2, J_3) . The latter is composed of

$$I_1 = \sigma_{ii} = 3\sigma_m \quad \text{first stress invariant,}$$

$$J_2 = \frac{1}{2} s_{ij} s_{ij} \quad \text{second deviatoric stress invariant, and}$$

$$J_3 = \det s_{ij} \quad \text{third deviatoric stress invariant,}$$

where σ_{ij} and s_{ij} are the stress and deviatoric stress tensors expressed in index notation, respectively, and σ_m is the mean stress. Note that these stress quantities are related via

$$\sigma_{ij} = s_{ij} + \sigma_m \delta_{ij}$$

where δ_{ij} is the Kronecker delta. A much more convenient and geometrically intuitive way to express a stress state is by (I_1, J_2, θ) or $(\sigma_o, \tau_o, \theta)$, where

$$\theta = \frac{1}{3} \arccos \left(\frac{3\sqrt{3} J_3}{2(J_2)^{\frac{3}{2}}} \right)$$

is the angle of similarity or Lode angle defined on a deviatoric stress plane (i.e., any plane orthogonal to the hydrostatic axis) such that $0 \leq \theta \leq \frac{\pi}{3}$, and the octahedral normal and shear stresses are defined as

$$\sigma_o = \frac{1}{3} I_1 = \sigma_m \quad \text{and} \quad \tau_o = \sqrt{\frac{2}{3} J_2}$$

respectively. The combination of octahedral stresses and angle of similarity $(\sigma_o, \tau_o, \theta)$ for expressing any stress state is heavily used in research that reports experimental data of concrete.

- Similar expressions to the ones presented for the octahedral stresses can be derived for the volumetric strain (ε_o) and the deviatoric strain (γ_o) .
- The stress-strain relations that describe the nonlinear deformational behavior of concrete are derived from curves that best fit the experimental data portrayed in $\sigma_o - \varepsilon_o$, $\tau_o - \gamma_o$ and

$\tau_o - \varepsilon_o$ plots. Due to the nature of the data, the relations to be presented are only valid when the stress units employed are N/mm² or MPa. It is important to highlight that the experimental data indicate that when subjecting concrete to hydrostatic pressure (σ_o) the deformation consists of mainly volumetric strain (ε_o), while when subjecting concrete to deviatoric stresses (τ_o), after applying a certain level of hydrostatic stress, the deformation consists of both deviatoric and volumetric strains (γ_o, ε_o). The empirical stress-strain relations capture these phenomena and at any given time are defined by

$$\sigma_o = 3K_s (\varepsilon_o - \varepsilon^{TH}) - \sigma_{id}$$

and

$$\tau_o = 2G_s \gamma_o,$$

where ε^{TH} is the thermal strain defined as in Section 3.1.4, K_s and G_s are the secant bulk and shear moduli, respectively, and σ_{id} is termed “internal hydrostatic stress due to an external deviatoric stress”. The secant bulk and shear moduli are defined as

$$K_s = \begin{cases} \frac{K_c}{1 + A \left(\frac{-\sigma_o}{f_c} \right)} & \text{for } \frac{-\sigma_o}{f_c} \leq 2 \\ \frac{K_c}{1 + 2^{b-1} Ab + 2^b (b-1) + A \left(\frac{\sigma_o}{f_c} \right)^{-1}} & \text{for } \frac{-\sigma_o}{f_c} > 2 \end{cases}$$

and

$$G_s = \frac{G_c}{1 + C \left(\frac{\tau_o}{f_c} \right)^{d-1}},$$

respectively, and

$$\sigma_{id} = \frac{k f_c}{1 + l \left(\frac{-\sigma_o}{f_c} \right)^m} \left(\frac{\tau_o}{f_c} \right)^n.$$

The following material parameters are used: K_e , G_e , A , b , C , d , k , l , m and n . These parameters are functions f_c and have been derived for values of f_c in the range of 16–65 N/mm². Outside of this range, all of these parameters remain constant and equal to their values for either $f_c = 16$ N/mm² or $f_c = 65$ N/mm². The expressions of these material parameters are

$$K_e = 11000 + 3.2 f_c^2,$$

$$G_e = 9224 + 136 f_c + 3296 \cdot 10^{-15} f_c^{8.273},$$

$$A = \begin{cases} 0.516 & \text{for } f_c \leq 31.7 \text{ N/mm}^2, \\ \frac{0.516}{1 + 0.0027 (f_c - 31.7)^{2.397}} & \text{for } f_c > 31.7 \text{ N/mm}^2, \end{cases}$$

$$b = 2 + 1.81 \times 10^{-8} f_c^{4.461},$$

$$C = \begin{cases} 3.573 & \text{for } f_c \leq 31.7 \text{ N/mm}^2, \\ \frac{3.573}{1 + 0.0134 (f_c - 31.7)^{1.414}} & \text{for } f_c > 31.7 \text{ N/mm}^2, \end{cases}$$

$$d = \begin{cases} 2.12 + 0.0183 f_c & \text{for } f_c \leq 31.7 \text{ N/mm}^2, \\ 2.7 & \text{for } f_c > 31.7 \text{ N/mm}^2, \end{cases}$$

$$k = \frac{4}{1 + 1.087 (f_c - 15)^{0.23}},$$

$$m = \begin{cases} -2.415 & \text{for } f_c \leq 31.7 \text{ N/mm}^2, \\ -3.531 + 0.0352 f_c & \text{for } f_c > 31.7 \text{ N/mm}^2, \end{cases}$$

$$n = \begin{cases} 1 & \text{for } f_c \leq 31.7 \text{ N/mm}^2, \\ 0.3124 + 0.0217 f_c & \text{for } f_c > 31.7 \text{ N/mm}^2. \end{cases}$$

As one can clearly notice, these expressions are unit dependent. Furthermore, K_e and G_e , are the elastic bulk and shear moduli, N/mm^2 , while the remaining material parameters do not have units since they are either coefficients or exponents.

- The stress-strain relations that describe concrete when it unloads from a given compressive stress level, reloads back to such a stress level, and is in tension are defined by

$$\sigma_o = 3K_e (\varepsilon_o - \varepsilon^{TH}) \quad \text{and} \quad \tau_o = 2G_e \gamma_o.$$

These expressions are the linear elastic constitutive equations written in terms of the octahedral stresses and strains, and governed by the corresponding elastic moduli of concrete.

- The criterion that identifies nonlinear compressive loading from the other possible linear loading states is dictated by comparing the octahedral shear stress between two time steps such that nonlinear loading occurs when ${}^{t+\Delta t}\tau_o > {}^t\tau_o^{\max}$, where ${}^t\tau_o^{\max}$ is the maximum octahedral shear stress (during nonlinear loading) up to time t .

- In some analyses the temperature effects can be significant and, therefore, the thermal strains need to be included. These strains (ε^{TH}) are taken into account by assuming that the material is either an isotropic temperature independent or dependent material (see Section 3.1.4). If the former is assumed, then the thermal strains are calculated following Eq. (3.1-16), while if the latter applies, then the thermal strains are given by Eq. (3.1-17).

When the material is isotropic temperature dependent, the following material properties can be defined as temperature dependent: uniaxial cylinder compressive strength (f_c), mean

coefficient of thermal expansion (α) and fracture energy (see the description of the tension softening option below). Note that when temperature effects are included, ADINA will require temperature data as input (see Section 5.6).

- The AUI command that defines the data fitted concrete material model in ADINA is MATERIAL DF-CONCRETE. In order to adapt any model to the above constitutive equations, this command provides the parameter FC, which defines the uniaxial cylinder compressive strength of the user's concrete, and the parameter CONFAC, which converts FC from the user's stress unit to N/mm² such that $16 \text{ N/mm}^2 \leq \text{FC} \times \text{CONFAC} \leq 65 \text{ N/mm}^2$.

When temperature effects are desired, the parameter ALPHA, which defines the mean coefficient of thermal expansion, must be specified.

- The failure surface that defines the ultimate strength values of concrete is derived from curves that best fit the experimental data portrayed in $\tau_o/f_c - \sigma_o/f_c$ plots. These data is usually presented on meridian planes (i.e., any plane that contains the hydrostatic axis and is defined by a constant angle of similarity) as well as on deviatoric planes (i.e., any plane that is perpendicular to the hydrostatic axis and is defined by a constant octahedral normal stress). Again, due to the nature of the data, most of the relations that follow are only valid when the stress units employed are N/mm² or MPa. The failure surface function employed by the data fitted concrete material model is given by

$$g = \tau_o - \frac{0.944 f_c}{r} \left(\text{TP} - \frac{\sigma_o}{f_c} \right)^{0.724} - \text{SP} f_c = 0,$$

where r is an elliptic function, TP is a percentage of f_c that represents the amount of tensile hydrostatic stress allowed by g such that $0.05 \leq \text{TP} \leq 0.1$, and SP is a percentage of f_c that represents the amount of additional octahedral shear stress allowed by g such that $0.0 \leq \text{SP} \leq 0.05$. Note that if $g > 0$, then the stress state of concrete is “outside” the failure surface and it indicates that failure has occurred in the form of a crack or crush. The elliptic function is defined as

$$r = \frac{4(1-e^2)\cos^2\theta + (2e-1)^2}{2(1-e^2)\cos\theta + (2e-1)\sqrt{4(1-e^2)\cos^2\theta + 5e^2 - 4e}},$$

where $0 \leq \theta \leq \frac{\pi}{3}$ and e denotes the eccentricity of the failure surface on a deviatoric plane. The data fitted concrete material model defines the eccentricity as

$$e = 0.670551 \left(\text{TP} - \frac{\sigma_o}{f_o} \right)^{0.133},$$

which, in essence, is the ratio of radial distances to the cross section of the failure surface (on a deviatoric plane) at $\theta = 0$ and $\theta = \frac{\pi}{3}$.

A description of the elliptic function can be found in

ref. Ph. Menétrey and K. J. Willam, “Triaxial Failure Criterion for Concrete and Its Generalization” *ACI Structural Journal*, 92(3), pp. 311-318, 1995.

Basically, the elliptic function defines the cross section of the failure surface that best fits the experimental data. Moreover, this failure surface portrays a somewhat triangular cross section for tensile and small compressive octahedral normal stresses, which tends to change to a somewhat circular cross section for larger compressive octahedral normal stresses (see Fig. 3.7-9).

- It is important to highlight that both TP and SP can be user input and are parameters of the AUI command MATERIAL DF-CONCRETE. Basically, they provide some flexibility to the failure surface by “stretching” it in different directions. The former parameter “stretches” it along the tensile side of the hydrostatic axis, while the latter “dilates” it on the deviatoric plane.
- The smeared crack approach, which governs part of the post-failure behavior of the data fitted concrete material model follows a “non-orthogonal fixed crack” type of modeling, that is, one that defines the crack planes of an integration point when they first

occur such that they stay valid for the remaining of the simulation and where the crack planes are not necessarily orthogonal to each other. The modeling of a crack starts after cracking is detected by the failure surface function and hinges on the idea that concrete retains its strength only on the defined crack planes, which indicates a brittle type of cracking. Consequently, the stress-strain relations for cracking are understood the best when they are presented in the local coordinate system (i.e., the coordinate system of the crack of interest) and always need to be transformed to the global coordinate system. Following an incremental formulation, these are

$$\Delta \boldsymbol{\sigma}^{(i)} = \mathbf{C}^{(i)} \Delta \boldsymbol{\varepsilon}^{(i)},$$

where $\Delta \boldsymbol{\sigma}^{(i)}$, $\mathbf{C}^{(i)}$ and $\Delta \boldsymbol{\varepsilon}^{(i)}$ denote the stress increment, constitutive matrix and engineering strain increment in the coordinate system corresponding to the i -th crack, respectively.

- An integration point of a 3-D solid element can take up to three cracks. The first crack plane is defined to be orthogonal to the direction of the maximum tensile principal stress existing prior to failure. For notation purposes, the first axis of the coordinate system of the first crack is assumed to be aligned with the direction of such a stress. After failure, this stress is set to zero (see Figure 3.7-10) and the corresponding constitutive matrix is defined as

$$\mathbf{C}^{(1)} = \begin{pmatrix} \text{STIFAC } M & 0 & 0 & 0 & 0 & 0 \\ 0 & M & N & 0 & 0 & 0 \\ 0 & N & M & 0 & 0 & 0 \\ 0 & 0 & 0 & \text{SHEFAC } G & 0 & 0 \\ 0 & 0 & 0 & 0 & \text{SHEFAC } G & 0 \\ 0 & 0 & 0 & 0 & 0 & G \end{pmatrix}$$

where STIFAC and SHEFAC are the normal and shear stiffness reduction factors, respectively, such that $0.0 < \text{STIFAC} < 1.0$ and $0.0 < \text{SHEFAC} < 1.0$, and

$$M = K_t + \frac{4}{3}G_t, \quad N = K_t - \frac{2}{3}G_t \quad \text{and} \quad G = G_t,$$

where K_t and G_t are the tangent bulk and shear moduli prior to failure, respectively. When the state prior to failure is considered to be a nonlinear compressive state, these moduli are defined as

$$K_t = \begin{cases} \frac{K_e}{1 + Ab \left(\frac{-\sigma_o}{f_c} \right)^{b-1}} & \text{for } \frac{-\sigma_o}{f_c} \leq 2 \\ \frac{K_e}{1 + 2^{b-1} Ab} & \text{for } \frac{-\sigma_o}{f_c} > 2 \end{cases}$$

and

$$G_t = \frac{G_e}{1 + Cd \left(\frac{\tau_o}{f_c} \right)^{d-1}},$$

otherwise they are defined as their elastic moduli counterparts. Note that the material parameters and terms of the last two relations have been defined previously.

An integration point presenting a single crack can in a future time step remain in the one-crack state, which indicates that the crack stays open, or fail again, that is, either follow the constitutive relations of the second crack (see next bullet) or crush, or return to its previous state, which indicates that the crack closes.

- A second crack forms when under a one-crack state, failure is again detected by the failure surface function such that there is at least one tensile principal stress. The second crack plane is defined to be orthogonal to the direction of the current maximum tensile principal stress. Note that the first and second crack planes need not be perpendicular to each other and this fact conveys that concrete retains its strength only in the direction defined by the intersection of these two crack planes. For notation purposes, the first axis of the coordinate system of the second crack is assumed to be aligned with the direction of the current maximum tensile

principal stress and the second axis is aligned with the intersection of the two crack planes. After failure, the stresses along the first and third axes are set to zero (see Fig. 3.7-11) and the corresponding constitutive matrix is defined as

$$\mathbf{C}^{(2)} = \begin{pmatrix} \text{STIFAC } M & 0 & 0 & 0 & 0 & 0 \\ 0 & M & 0 & 0 & 0 & 0 \\ 0 & 0 & \text{STIFAC } M & 0 & 0 & 0 \\ 0 & 0 & 0 & \text{SHEFAC } G & 0 & 0 \\ 0 & 0 & 0 & 0 & \text{SHEFAC } G & 0 \\ 0 & 0 & 0 & 0 & 0 & \text{SHEFAC } G \end{pmatrix}$$

An integration point presenting two cracks can in a future time step remain in the two-crack state, which indicates that both cracks stay open, or fail again, that is, either follow the constitutive relations of the third crack (see next bullet) or crush, or return to the one-crack state, which indicates that the second crack closes while the first crack remains open. Note that in the last case, once in the one-crack state, the integration point can return to its state prior to cracking within the same time step, which indicates that both cracks close.

- A third and last possible crack forms when under a two-crack state failure is detected one more time by the failure surface function such that the stress along the intersection of the two previous crack planes is tensile. This is a fully cracked state that resembles an almost complete loss of load-carrying capacity. Note that the coordinate system of the third crack is the same as that of the second crack, and after failure the stresses along the axes of such coordinate system are set to zero (see Fig. 3.7-12) and the corresponding constitutive matrix is defined as

$$\mathbf{C}^{(3)} = \begin{pmatrix} \text{STIFAC } M & 0 & 0 & 0 & 0 & 0 \\ 0 & \text{STIFAC } M & 0 & 0 & 0 & 0 \\ 0 & 0 & \text{STIFAC } M & 0 & 0 & 0 \\ 0 & 0 & 0 & \text{SHEFAC } G & 0 & 0 \\ 0 & 0 & 0 & 0 & \text{SHEFAC } G & 0 \\ 0 & 0 & 0 & 0 & 0 & \text{SHEFAC } G \end{pmatrix}$$

An integration point presenting three cracks can in a future time step remain in the fully cracked state, which indicates that all three cracks stay open, or return to the two-crack state, which indicates that the third crack closes while the first and second crack stay open. An integration point that is fully cracked cannot fail again. Moreover, given the right strain conditions (see next bullet) such an integration point can return within the same time step to the one-crack state or even to its state prior to cracking.

- The criterion that identifies when a crack can be closed is dictated by comparing strain values along the direction of the stress that caused the formation of the crack. For the one- and two-crack cases, the direction is perpendicular to their corresponding crack planes, while for the fully cracked state it is along the line that defines the intersection of these two crack planes. In order to close a crack, the current strain value must be compressive and smaller than the strain value present when the crack was formed.

An integration point presenting a closed crack can in a future time step remain with the closed crack or fail again. When failure is detected by the failure surface function, the corresponding integration point can either reopen the closed crack or crush. Note that cracks are assumed to be history dependent and, therefore, a reopened crack will have the orientation defined when it was initially opened..

- Note that both STIFAC and SHEFAC can be user input and are parameters of the AUI command MATERIAL DF-CONCRETE. The values assigned to these factors are constant for the entire duration of the simulation and must be chosen carefully by the user.

In essence, these factors dictate how the concrete's stiffness degrades during cracking. On the one hand they intend to capture what occurs during cracking, while on the other hand they help avoid possible numerical difficulties associated with the degradation of the stiffness matrix (to a singular matrix), especially when a large number of cracks start forming. The normal stiffness reduction factor must reflect the brittle nature of cracking, that is, the loss of strength outside the crack plane. It is recommended that STIFAC be a very small (positive) value, i.e., $\text{STIFAC} \ll 1.0$. The shear stiffness reduction factor is related to the notion that aggregate interlock occurs on the formed crack planes due to the roughness present on the separating surfaces of a crack. An adequate value for it must neither be too big that creates an unrealistically large stiffness nor too small that cause numerical instability. It is recommended that SHEFAC be within $0.1 \leq \text{SHEFAC} \leq 0.5$.

- The other possible post-failure behavior of the data fitted concrete material model is crushing, which portrays the brittle nature of concrete in compression. After failure is detected by the failure surface function, crushing occurs when all the principal stresses are compressive. Crushing is modeled as a complete and permanent loss of load-carrying capacity and, therefore, all the stress components are set to zero (see Fig. 3.7-13). For numerical reasons, the corresponding constitutive matrix is defined as the one for the fully cracked state but in the global coordinate system.

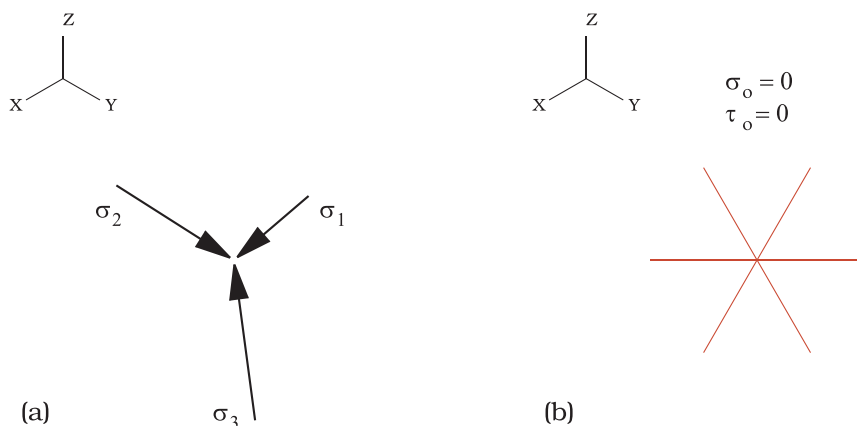


Figure 3.7-13: Crush modeling of the data fitted concrete material

model (a) before, and (b) after

- An optional feature of the data fitted concrete material model is the inclusion of tension softening after cracking. The tension softening behavior is assumed to follow a linear stress-strain curve (see Fig. 3.7-14) and, therefore, will replace the default brittle cracking behavior. Tension softening is possible only for the one- and two-crack states. In the one-crack state tension softening applies to $\sigma_x^{(1)}$, while in the two-crack state both $\sigma_x^{(2)}$ and $\sigma_z^{(1)}$ will follow the tension softening behavior. In addition, the corresponding components of the constitutive matrices $\mathbf{C}^{(1)}$ and $\mathbf{C}^{(2)}$ of the one- and two-crack states, respectively, will be modified to account for tension softening.

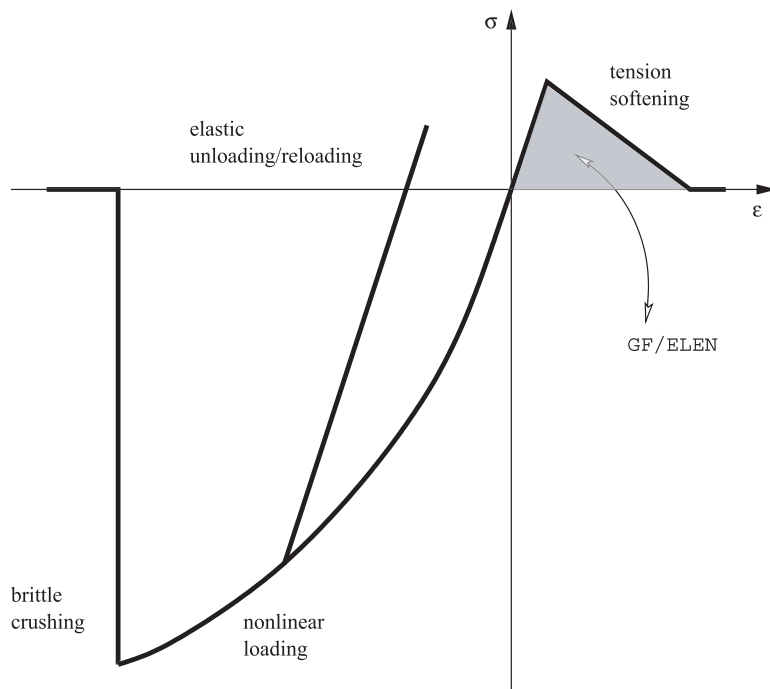


Figure 3.7-14: Stress-strain behavior of the data fitted concrete material model

- The tension softening behavior is defined by the fracture energy

value and the characteristic length of the concrete model so as to ensure a numerical solution that is not dependent on the concrete's finite element mesh. The former is defined as the area under the tensile stress-displacement curve and via the parameter of the AUI command MATERIAL DF-CONCRETE. The latter can be usually taken as the length of the element along the direction perpendicular to the first crack plane and is defined by the parameter ELEN MATERIAL DF-CONCRETE. If the parameter ELEN is not input by the user, then ADINA will define the characteristic length of the model as the largest length of the element. However, this is not always desirable and, therefore, it is recommended that the user input a value for ELEN. Refer to Fig. 3.7-14 for a graphical interpretation of the relationship between GF and ELEN.

Note that since tension softening replaces the default brittle cracking behavior, the value of the parameter STIFAC is controlled by ADINA.

- The post-processing of the simulation results of the data fitted concrete material model is done via the AUI. For each integration point of an element, this material model outputs the following quantities (where their corresponding AUI variable names are enclosed in parentheses):

- ▶ (STRESS-(XYZ));
- ▶ engineering total strains (STRAIN-(XYZ));
- ▶ thermal strain (THERMAL_STRAIN);
- ▶ temperature (ELEMENT_TEMPERATURE);
- ▶ failure surface function value (FAILURE_SURFACE_FUNCTION);
- ▶ data fitted concrete material state flag (DF-CONCRETE_FLAG);
- ▶ crack state flag (CRACK_FLAG);
- ▶ normal directions of the crack planes (SIGMA-P1_ANGLE and FE_SIGMA-P2_ANGLE for 2-D

solid elements, FE_SIGMA-P1_DIRECTION-(XYZ), FE_SIGMA-P2_DIRECTION-(XYZ) and FE_SIGMA-P3_DIRECTION-(XYZ)} for 3-D solid elements);

- ▶ stress values at the time of crack formation along the normal directions (SIGMA-P1, FE_SIGMA-P2 and FE_SIGMA-P3); and
- ▶ strain values at the time of crack formation along the normal directions (FE_STRAIN-P1, FE_STRAIN-P2 and FE_STRAIN-P3).

For additional simulation results available for post-processing, refer to Section 13.1.1.

- A useful way of post-processing the results associated with cracking and crushing is by using the vector plot capabilities of the AUI. The post-failure-related element vectors applicable to the data fitted concrete material model, which are displayed with the AUI command `EVECTORPLOT`, are:

- ▶ `OPEN_CRACKS`;
- ▶ `CLOSED_CRACKS`;
- ▶ `CRUSHED`;
- ▶ `CRACKS`;
- ▶ `CRACK_STRESS`; and
- ▶ `CRACK_STRAIN`.

Although the above variables have self-explanatory names, it should be noted that `CRACKS` displays the open and closed crack planes as well as the crushed data, `CRACK_STRESS` and `CRACK_STRAIN` display the stress and strain values at the time of crack formation (see previous bullet), respectively. For additional information regarding vector plotting, refer to Section 5.5 of the AUI Command Reference Manual, Volume IV: Display

Processing.

- The following is a list of modeling hints and suggestions for simulations of concrete structures that will use the data fitted concrete material model.

1. *The data fitted concrete material model is intended for modeling reinforced concrete structures.* Although unreinforced or plain concrete structures can be modeled with this material model by using the tension softening option, the aim of this model is to simulate concrete structures that contain proper reinforcement in the applications of monotonic and/or dynamic loading.

2. Use of the data fitted concrete material model is recommended when the available material parameters of concrete cannot fully describe complex stress-strain laws and/or failure surfaces. The advantage of this particular concrete material model is that it is fully described by a single material parameter, which is the uniaxial cylinder compressive strength.

3. As stated above, the inclusion of proper reinforcement is highly desirable in concrete structures that are simulated using the data fitted concrete material model. The reason behind this statement is because this model treats concrete as a brittle material when it cracks and crushes. Consequently, plain concrete structures that use the data fitted concrete material model with its default brittle behavior will, in general, fail with somewhat smaller loads than if they were to be simulated with a concrete material model that includes, for example, softening.

4. The values of the parameters TP or SP can be increased to overcome convergence difficulties. In particular, increase TP for structures that show large tensile normal octahedral stresses, e.g., reinforced concrete beams that tend to flex, and increase SP for structures that show large shear octahedral stresses, e.g., a structure lacking shear reinforcement.

5. The values of the parameters STIFAC or SHEFAC can be modified to overcome convergence difficulties. In particular, keep in mind that $STIFAC \ll 1.0$ and $0.1 \leq SHEFAC \leq 0.5$. The former parameter is intended to model brittle cracking,

while the latter models the aggregate interlock phenomenon due to cracking.

In the case of plain concrete structures that are simulated with the data fitted concrete material model, the above recommendation for the parameter SHEFAC can provide undesirable results. It is instead recommended that $\text{SHEFAC} \ll 1.0$.

6. The geometry, type of loading, boundary conditions, mesh size and pattern, and analysis type are other modeling criteria that must be carefully selected by the user when simulating concrete structures.

For instance, choosing between force or displacement loading can affect how far a simulation can be accurately solved. Also, 8-, 20- or 27-node 3-D solid elements; or fine or coarse meshes can affect the performance of a simulation. The choice of element type usually depends on the type of deformation the element is expected to sustain during the simulation. Regarding the mesh size, it is usually recommended that it be neither too fine that creates localized failure zones nor too coarse that gives unrealistic results.

The type of analysis used for simulating concrete structures can affect the quality of the final results. This modeling criterion plays an important role when a concrete structure starts presenting convergence difficulties due to cracking and/or crushing, that is, due to the degradation of the stiffness matrix of the concrete structure. ADINA has a number of numerical methods that can help obtain accurate results for nonlinear static analyses (see Section 7.2). For example, in the case of a static analysis that presents convergence difficulties, adding line search to improve the iterative solution procedure, or changing the analysis to a low speed dynamic analysis, or simply using the automatic-time-stepping (ATS) method, or alternatively changing the analysis to use the load-displacement-control (LDC) method can help overcome such numerical problems. As a general rule of thumb, it is recommended to use the ATS method with or without line search to overcome the unavoidable numerical difficulties present in concrete modeling.

3.7.3 Creep/Shrinkage effects

- Creep/Shrinkage effects can be included in the DF-concrete(data fitted concrete) material model.

- Creep/Shrinkage models are adopted from ACI 209R-92 code and CEB FIP 1990 code, see the following references:

ref. ACI 209R-92, 2008. Modeling and Calculating Shrinkage and Creep in Hardened Concrete.

ref. CEP-FIP model code 1993. Design of Concrete Structures.

ref. Bazant, Z.P. “Criteria for Rational Prediction of Creep and Shrinkage of Concrete”, *Revue Francaise de Genie Civil* 2 (3-4), 1999, 61-89.

- The ACI209R-92 creep/shrinkage model is as follows:

Creep strain coefficient:

$$\varphi(t, t_0) = \frac{(t - t_0)^w}{d + (t - t_0)^w} \cdot (2.35 \cdot \gamma_c)$$

$$\gamma_c = \gamma_{c,t_0} \cdot \gamma_{c,RH} \cdot \gamma_{c,VS} \cdot \gamma_{c,S} \cdot \gamma_{c,\psi} \cdot \gamma_{c,\alpha}$$

$$\gamma_{c,t_0} = 1.25 \cdot t_0^{-0.118} \text{ for moist curing}$$

$$\gamma_{c,t_0} = 1.13 \cdot t_0^{-0.094} \text{ for steam curing}$$

t_0 is the age of concrete at loading

$$\gamma_{c,RH} = 1.27 - 0.67 \cdot h \text{ for } h \geq 0.4, \text{ } h \text{ is relative humidity}$$

$$\begin{aligned} \gamma_{c,VS} &= \frac{2}{3} (1 + 1.13 \cdot e^{\{-0.0213(VS)\}}), \text{ in mm} \\ &= \frac{2}{3} (1 + 1.13 \cdot e^{\{-0.54(VS)\}}), \text{ in inch} \end{aligned}$$

VS is the volume-surface ratio

$$\begin{aligned}\text{slump factor, } \gamma_{C,S} &= 0.82 + 0.00264 \cdot s(mm) \\ &= 0.82 + 0.067 \cdot s(inch)\end{aligned}$$

$$\text{fine aggregate factor, } \gamma_{C,\psi} = 0.88 + 0.0024 \cdot \psi(\%)$$

$$\text{air content factor, } \gamma_{C,\alpha} = 0.46 + 0.09 \cdot \alpha(\%) \geq 1$$

$$\text{with } d=10, \text{ and } \psi=0.6$$

Shrinkage strain:

$$\varepsilon_{sh}(t, t_s) = \frac{(t - t_s)^\alpha}{f + (t - t_s)^\alpha} \cdot 780 \times 10^{-6} \cdot \gamma_{sh}$$

$$\gamma_{sh} = \gamma_{sh,t_s} \cdot \gamma_{sh,RH} \cdot \gamma_{sh,vs} \cdot \gamma_{sh,s} \cdot \gamma_{sh,\psi} \cdot \gamma_{sh,c} \cdot \gamma_{sh,\alpha}$$

$$\begin{aligned}\gamma_{sh,t_s} &= 1.202 - 0.2337 \cdot \log(t_s) \text{ for moist curing} \\ &= 1 \text{ for steam curing}\end{aligned}$$

t_s is the age of concrete at the beginning of shrinkage

$$\begin{aligned}\gamma_{sh,RH} &= 1.40 - 1.02 \cdot h \text{ for } 0.4 \leq h \leq 0.8 \\ &= 3.00 - 3.0 \cdot h \text{ for } 0.8 \leq h \leq 1\end{aligned}$$

$$\begin{aligned}\gamma_{sh,vs} &= 1.2 \cdot e^{\{-0.00472(VS)\}}, \text{ in } mm \\ &= 1.2 \cdot e^{\{-0.12(VS)\}}, \text{ in } inch\end{aligned}$$

$$\begin{aligned}\text{slump factor, } \gamma_{sh,s} &= 0.89 + 0.00161 \cdot s(mm) \\ &= 0.89 + 0.041 \cdot s(inch)\end{aligned}$$

$$\begin{aligned} \text{fine aggregate factor, } \gamma_{sh,\psi} &= 0.30 + 0.014 \cdot \psi(\%), \text{ for } \psi \leq 50 \\ &= 0.90 + 0.002 \cdot \psi(\%), \text{ for } \psi > 50 \end{aligned}$$

$$\begin{aligned} \text{cement content factor, } \gamma_{sh,c} &= 0.75 + 0.00061 \cdot c(\text{kg/m}^3) \\ &= 0.75 + 0.00036 \cdot c(\text{lb/yd}^3) \end{aligned}$$

$$\text{air content factor, } \gamma_{sh,\alpha} = 0.95 + 0.008 \cdot \alpha(\%) \geq 1$$

- The CEB-FIP 1990 creep/shrinkage model is as follows:

Creep strain coefficient:

$$\varphi(t, t_0) = \varphi_h \cdot \beta(f_c) \cdot \frac{1}{0.1 + (t_0)^{0.2}} \cdot \left[\frac{(t - t_0)}{\beta_h + (t - t_0)} \right]^{0.3}$$

$$\varphi_h = \left[1 + \frac{1 - h/100}{0.1 \cdot \sqrt[3]{L}} \right]$$

$$\beta(f_c) = \frac{16.8}{\sqrt{f_c}}$$

$$\beta_h = 1.5 \cdot L \cdot \left[1 + (1.2 \cdot h/100)^{18} \right] + 250$$

$$L = 2A_c / u$$

t_0 is the age of concrete at loading

h is the relative humidity of the environment (%)

A_c is the cross section area and u is the perimeter

f_c is the mean compressive strength at 28 days

Shrinkage strain:

$$\varepsilon_{sh} = [160 + 10 \cdot \beta_{sc} \cdot (9 - f_c/10)] \cdot 10^{-6} \cdot \beta_h \cdot \left[\frac{t - t_s}{350 \cdot (L/100)^2 + t - t_s} \right]^{0.5}$$

$$\begin{aligned}\beta_h &= -1.55 \cdot [1 - (h/100)^3] \text{ for } 40 \leq h(\%) < 99 \\ &= +0.25 \text{ for } h(\%) \geq 99 \\ \beta_{sc} &= 4 \text{ for slowly hardening cements SL} \\ &= 5 \text{ for normal or rapid hardening cements N and R} \\ &= 8 \text{ for rapid hardening high strength cements RS}\end{aligned}$$

f_c is the mean compressive of concrete at 28 days

t_s is the concrete age at the beginning of shrinkage

- The ADINA creep strain equation is

$$\begin{aligned}\bar{\varepsilon}^{cr} &= \frac{\bar{\sigma}}{E(t)} \cdot \left[\frac{a_1 + t^{a_2}}{a_3 + t^{a_4}} \right]^{a_{13}} \cdot \left[\frac{a_5 + t_0^{a_6}}{a_7 + t_0^{a_8}} \right]^{a_{14}} \cdot \left[\frac{a_9 + (t - t_0)^{a_{10}}}{a_{11} + (t - t_0)^{a_{12}}} \right]^{a_{15}} \\ E(t) &= a_{22} \exp[a_{23}(1 - (a_{24}/t)^{a_{25}})]\end{aligned}$$

- The ADINA shrinkage strain equation is

$$\varepsilon^{sh} = \left[\frac{a_{26} + (t - t_s)^{a_{27}}}{a_{28} + (t - t_s)^{a_{29}}} \right]^{a_{30}} \cdot a_{31}$$

- Calculation of the creep strain, using creep multiplier γ

$$\begin{aligned}{}^{t+\Delta t}\gamma &= \frac{3}{2} \cdot \frac{{}^{t+\Delta t}\dot{\bar{\varepsilon}}^{cr}}{{}^{t+\Delta t}\bar{\sigma}} \\ \Delta \bar{\varepsilon}_{ij}^{cr} &= \Delta t \cdot {}^{t+\Delta t}\gamma \cdot {}^{t+\Delta t}S_{ij}\end{aligned}$$

- Shrinkage strain is calculated simply as the thermal strain
- The equation of total strain is

$$\varepsilon = \varepsilon^{co} + \varepsilon^{th} + \varepsilon^{cr} + \varepsilon^{sh}$$

where

ε^{co} is the concrete strain

ε^{th} is the thermal strain

ε^{cr} is the creep strain

ε^{sh} is the shrinkage strain

3.8 Rubber and foam material models

- The rubber and foam material models in ADINA are the **Mooney-Rivlin**, **Ogden**, **Arruda-Boyce**, **hyper-foam**, **Sussman-Bathe** and **eight-chain** material models.
- These material models can be employed with the **2-D solid**, **3-D solid** and **3D-shell** elements.
- These material models can be used with the **large displacement/large strain** formulation. A TL formulation is employed.
- These material models include the following effects:
 - ▶ isotropic effects (required), see Section 3.8.1
 - ▶ viscoelastic effects using the Holzapfel model (optional), see Section 3.8.2
 - ▶ viscoelastic effects using the Bergström-Boyce model (optional), see Section 3.8.2
 - ▶ Mullins effects (optional), see Section 3.8.3
 - ▶ orthotropic effects (optional), see Section 3.8.4
 - ▶ thermal strain effects (optional), see Section 3.8.5
- These material models allow three types of temperature dependence:
 - ▶ no temperature dependence
 - ▶ time-temperature superposition for viscoelastic effects, see Section 3.8.6. This type of material is referred to as a thermorheologically simple material (TRS material).

- ▶ full temperature dependence, see Section 3.8.7.

Table 3.8-1 shows a summary of the features.

Table 3.8-1: Summary of features for rubber and foam material models

	Viscoelastic effects, Holzapfel model ¹	Viscoelastic effects, Bergström-Boyce model ^{1,2}	Mullins effects ¹	Orthotropic effects	Thermal strain effects
No temperature dependence	yes	yes	yes	yes	no
TRS temperature dependence	yes	no	yes	yes	yes
Full temperature dependence	yes	yes	yes	yes	yes

- 1) Viscoelastic effects and Mullins effects cannot be used together.
- 2) Orthotropic effects cannot be used with the Bergström-Boyce model

- When the material temperature dependence is TRS or full, heat generation from the rubber-like materials is included in TMC (thermo-mechanical-coupling) analysis.

3.8.1 Isotropic hyperelastic effects

- The rubber and foam material models include the following models for isotropic hyperelastic effects:
 - ▶ Mooney-Rivlin
 - ▶ Ogden
 - ▶ Arruda-Boyce
 - ▶ hyper-foam
 - ▶ Sussman-Bathe

► Eight-chain

- Only the Mooney-Rivlin, Ogden, Arruda-Boyce, Sussman-Bathe, and eight-chain material models are supported in explicit dynamic analysis.
- The isotropic hyperelastic effects are mathematically described by specifying the dependence of the strain energy density (per unit original volume) W on the Green-Lagrange strain tensor ε_{ij} .
- We now give a brief summary of the quantities and concepts used. For more information, refer to ref KJB, section 6.6.2. Here and below, we omit the usual left superscripts and subscripts for ease of writing. Unless otherwise stated, all quantities are evaluated at time t and referred to reference time 0.

Useful quantities are the Cauchy-Green deformation tensor C_{ij} , given by

$$C_{ij} = 2\varepsilon_{ij} + \delta_{ij} \quad (3.8-1)$$

where δ_{ij} is the Kronecker delta; the principal invariants of the Cauchy-Green deformation tensor,

$$I_1 = C_{kk}, \quad I_2 = \frac{1}{2}(I_1^2 - C_{ij}C_{ij}), \quad I_3 = \det \mathbf{C} \quad (3.8-2a,b,c)$$

the reduced invariants

$$J_1 = I_1 I_3^{-\frac{1}{3}}, \quad J_2 = I_2 I_3^{-\frac{2}{3}}, \quad J_3 = I_3^{\frac{1}{2}}, \quad (3.8-3a,b,c)$$

the stretches λ_i where the λ_i 's are the square roots of the principal stretches of the Cauchy-Green deformation tensor; and the reduced stretches

$$\lambda_i^* = \lambda_i (\lambda_1 \lambda_2 \lambda_3)^{-\frac{1}{3}} \quad (3.8-4)$$

Note that

$$J_3 = \lambda_1 \lambda_2 \lambda_3 \quad (3.8-5)$$

is the volume ratio (ratio of the deformed volume to the undeformed volume).

The strain energy density W is written either in terms of the invariants or in terms of the stretches. In many cases, the strain energy density is conveniently written as the sum of the deviatoric strain energy density W_D and the volumetric strain energy density W_V .

With knowledge of how the strain energy density W depends on the Green-Lagrange strain tensor (through the invariants or stretches), the 2nd Piola-Kirchhoff stress tensor is evaluated using

$$S_{ij} = \frac{1}{2} \left(\frac{\partial W}{\partial \epsilon_{ij}} + \frac{\partial W}{\partial \epsilon_{ji}} \right) \quad (3.8-6)$$

and the incremental material tensor is evaluated using

$$C_{ijrs} = \frac{1}{2} \left(\frac{\partial S_{ij}}{\partial \epsilon_{rs}} + \frac{\partial S_{ij}}{\partial \epsilon_{sr}} \right) \quad (3.8-7)$$

3.8.1.1 Mooney-Rivlin model

- The Mooney-Rivlin material model is based on the following expression:

$$\begin{aligned} W_D = & C_1 (I_1 - 3) + C_2 (I_2 - 3) + C_3 (I_1 - 3)^2 + C_4 (I_1 - 3)(I_2 - 3) + \\ & C_5 (I_2 - 3)^2 + C_6 (I_1 - 3)^3 + C_7 (I_1 - 3)^2 (I_2 - 3) + \\ & C_8 (I_1 - 3)(I_2 - 3)^2 + C_9 (I_2 - 3)^3 + D_1 \left(\exp(D_2 (I_1 - 3)) - 1 \right) \end{aligned} \quad (3.8-8)$$

where C_1 to C_9 and D_1 to D_2 are material constants.

- This strain energy density expression assumes a totally incompressible material ($I_3 = 1$) and is modified as explained below for plane strain, axisymmetric or 3-D analysis.

Plane stress analysis In plane stress analysis, the material is assumed to be totally incompressible. Therefore W_v is zero and $W = W_D$. A displacement-based finite element formulation is used, in which the incompressibility condition of the material is imposed by calculating the appropriate thickness of the material.

ref. KJB
Section 6.6.2

Plane strain, axisymmetric and 3-D analysis: In plane strain, axisymmetric and 3-D analysis, the material is modeled as compressible (that is, the bulk modulus is not infinite), but you can set the bulk modulus high so that the material is “almost incompressible”.

The Mooney-Rivlin strain energy density equation is modified by: 1) substituting for the invariants I_1, I_2 the reduced invariants J_1, J_2 , 2) removing the condition $I_3 = 1$, and 3) adding the volumetric strain energy density

$$W_v = \frac{1}{2} \kappa (J_3 - 1)^2 \quad (3.8-9)$$

where κ is the bulk modulus. This expression for the volumetric strain energy density yields the following relationship between the pressure and the volume ratio:

$$p = -\kappa (J_3 - 1) \quad (3.8-10)$$

The Mooney-Rivlin material model is intended for use when the bulk to shear modulus ratio is large. The large bulk-to-shear modulus ratio causes numerical difficulties for axisymmetric, plane strain and 3D elements. The mixed u/p formulation can be used (and is the default) for these elements. This formulation avoids the numerical difficulties by use of a separately interpolated pressure, as described in the following reference:

- ref. T. Sussman and K.J. Bathe, "A Finite Element Formulation for Nonlinear Incompressible Elastic and Inelastic Analysis," *J. Computers and Structures*, Vol. 26, No. 1/2, pp. 357-409, 1987.

The displacement-based formulation can also be used, but is not recommended.

Selection of material constants: The Mooney-Rivlin material description used here has constants C_1 to C_9 , constants D_1 to D_2 and the bulk modulus κ . Strictly speaking, this material law is termed a higher-order or generalized Mooney-Rivlin material law.

Choosing only $C_1 \neq 0$ yields the neo-Hookean material law, and choosing only $C_1 \neq 0$, $C_2 \neq 0$ yields the standard two-term Mooney-Rivlin material law. In ADINA, constants C_3 to C_9 can be chosen to more closely fit experimental data.

Constants D_1 and D_2 are primarily intended for modeling certain biological materials and need not be used otherwise.

The small strain shear modulus and small strain Young's modulus can be written in terms of these constants as (assuming $\kappa = \infty$)

$$G = 2[(C_1 + C_2) + D_1 D_2] \quad (3.8-11)$$

$$E = 6[(C_1 + C_2) + D_1 D_2] \quad (3.8-12)$$

These moduli must be greater than zero.

- The bulk modulus κ is used to model the compressibility of the material for plane strain, axisymmetric and 3-D analysis.
- ADINA assumes a default for the bulk modulus based on small strain near-incompressibility, i.e.,

$$\kappa = \frac{E}{3(1-2\nu)} \quad \text{with } \nu = 0.499 \quad (3.8-13)$$

where E is the small strain Young's modulus or, in terms of the small strain shear modulus G ,

$$\kappa = \frac{2G(1+\nu)}{3(1-2\nu)} = 500G \quad \text{for } \nu = 0.499 \quad (3.8-14)$$

This rule of thumb can be used to estimate the bulk modulus in the absence of experimental data. However, you can use lower values of the bulk modulus to model compressible materials.

- In explicit analysis, the program assumes the same bulk modulus based on small strain near-incompressibility. However, this can significantly reduce the stable time step. In such cases, is better to use a bulk modulus that results in $\nu = 0.49$.
- When explicit analysis with automatic time step calculation is used for a Mooney-Rivlin material, the critical time step is governed by the dilatational wave speed. This is most frequently an acceptable assumption since the material is almost incompressible.
- As the material deforms, the bulk to shear modulus ratio may change, because the instantaneous shear modulus is dependent on the amount of deformation. A value of the bulk modulus that corresponds to near incompressibility for small strains may not be large enough to correspond to near incompressibility for large strains.

3.8.1.2 Ogden material model

- The Ogden material model is based on the following expression:

$$W_D = \sum_{n=1}^9 \left(\frac{\mu_n}{\alpha_n} \left[\lambda_1^{\alpha_n} + \lambda_2^{\alpha_n} + \lambda_3^{\alpha_n} - 3 \right] \right) \quad (3.8-15)$$

where μ_n and α_n are the Ogden material constants.

- This strain energy density expression assumes a totally incompressible material ($I_3 = 1$) and is modified as explained below for plane strain, axisymmetric or 3-D analysis.

Plane stress analysis: The material is assumed to be totally

incompressible. Therefore W_V is zero and $W = W_D$. A displacement-based finite element formulation is used, exactly as for the Mooney-Rivlin material model described above.

ref. KJB
Section 6.6.2

Plane strain, axisymmetric and 3-D analysis: The material is modeled as compressible (that is, the bulk modulus is not infinite), but you can set the bulk modulus high so that the material is “almost incompressible”.

The Ogden strain energy density equation is modified by 1) substituting the reduced stretches for the corresponding stretches, 2) removing the condition $\lambda_1\lambda_2\lambda_3 = 1$, and 3) adding the volumetric strain energy density

$$W_V = \frac{1}{2}\kappa(\lambda_1\lambda_2\lambda_3 - 1)^2 = \frac{1}{2}\kappa(J_3 - 1)^2 \quad (3.8-16)$$

where κ is the bulk modulus. The relationship between the pressure and the volumetric ratio is the same as for the Mooney-Rivlin material description.

Either the u/p formulation (default) or the displacement-based formulation can be used. For comments about the u/p formulation, see the corresponding comments in the Mooney-Rivlin material description.

Selection of material constants: The Ogden material description used here has 19 constants: μ_n, α_n , $n = 1, \dots, 9$ and the bulk modulus. Choosing only $\mu_n, \alpha_n \neq 0$, $n = 1, 2, 3$ the standard three-term Ogden material description is recovered.

The small strain shear modulus and small strain Young's modulus can be written as (assuming $\kappa = \infty$)

$$G = \frac{1}{2} \sum_{n=1}^9 \mu_n \alpha_n \quad (3.8-17)$$

$$E = \frac{3}{2} \sum_{n=1}^9 \mu_n \alpha_n \quad (3.8-18)$$

These moduli must be greater than zero.

- When explicit analysis with automatic time step calculation is used for an Ogden material, the critical time step is governed by the dilatational wave speed. This is most frequently an acceptable assumption since the material is almost incompressible.
- The bulk modulus κ is used to model the compressibility of the material for plane strain, axisymmetric and 3-D analysis. For comments about the bulk modulus, see the corresponding comments about the bulk modulus in the Mooney-Rivlin material description.

3.8.1.3 Arruda-Boyce material model

- The Arruda-Boyce model is based on the following expression:

$$W_D = \mu \sum_{i=1}^5 \left[\frac{C_i}{\lambda_L^{2i-2}} (I_1^i - 3^i) \right] \quad (3.8-19)$$

in which $C_1 = \frac{1}{2}$, $C_2 = \frac{1}{20}$, $C_3 = \frac{11}{1050}$, $C_4 = \frac{19}{7000}$,

$C_5 = \frac{519}{673750}$ and in which there are two material constants: μ and λ_L .

- The Arruda-Boyce material model is described in the following reference:

ref. E.M. Arruda and M. C. Boyce, "A three-dimensional constitutive model for the large stretch behavior of rubber elastic materials", *J. Mech. Phys. Solids*, Vol. 41 (2), pp 389-412 (1993).

- The Arruda-Boyce material model is an approximation to the eight-chain material model described in Section 3.8.1.6.
- This strain energy density expression assumes a totally

incompressible material ($I_3 = 1$) and is modified as explained below for plane strain, axisymmetric or 3-D analysis.

Plane stress analysis: The material is assumed to be totally incompressible. Therefore W_V is zero and $W = W_D$. A displacement-based finite element formulation is used, exactly as for the Mooney-Rivlin material model described above.

ref. KJB
Section 6.6.2

Plane strain, axisymmetric and 3-D analysis: The material is modeled as compressible (that is, the bulk modulus is not infinite), but you can set the bulk modulus high so that the material is “almost incompressible”.

The Arruda-Boyce strain energy density equation is modified by: 1) substituting for the strain invariant I_1 the reduced strain invariant J_1 , 2) removing the condition $I_3 = 1$, and 3) adding the volumetric energy term

$$W_V = \kappa (J_3 \ln J_3 - (J_3 - 1)) \quad (3.8-20)$$

where κ is the small-strain bulk modulus. The relationship between the pressure and the volume ratio is

$$p = -\kappa \ln J_3 \quad (3.8-21)$$

Either the u/p formulation (default) or the displacement-based formulation can be used. For comments about the u/p formulation, see the corresponding comments in the Mooney-Rivlin material description.

When the u/p formulation is used, there should be at least one solution unknown. This is because the constraint equation used in the u/p formulation is nonlinear in the unknown pressures. Therefore equilibrium iterations are required for convergence, even when all of the displacements in the model are prescribed.

Selection of material constants:

Material constant μ is only an approximation to the initial shear

modulus. The relationship between material constant μ and the initial shear modulus μ_0 is

$$\mu_0 = \mu \left(1 + \frac{3}{5\lambda_L^2} + \frac{99}{175\lambda_L^4} + \frac{513}{875\lambda_L^6} + \frac{42039}{67375\lambda_L^8} \right)$$

3.8.1.4 Hyper-foam material model

- The hyper-foam model is based on the following expression:

$$W = \sum_{n=1}^N \frac{\mu_n}{\alpha_n} \left[\lambda_1^{\alpha_n} + \lambda_2^{\alpha_n} + \lambda_3^{\alpha_n} - 3 + \frac{1}{\beta_n} (J_3^{-\alpha_n \beta_n} - 1) \right] \quad (3.8-22)$$

in which there are the material constants $\mu_n, \alpha_n, \beta_n, n = 1, \dots, N$. The maximum value of N is 9.

- A material model similar to the hyper-foam material model is described in the following reference:

ref. B. Storåkers, "On material representation and constitutive branching in finite compressible elasticity", *J. Mech. Phys. Solids*, Vol. 34(2), pp 125-145 (1986).

In this reference, β_n is the same for all values of n .

- When it is required to compute the deviatoric and volumetric parts of the strain energy density, we use

$$W_D = \sum_{n=1}^N \frac{\mu_n}{\alpha_n} \left[\lambda_1^{\alpha_n} + \lambda_2^{\alpha_n} + \lambda_3^{\alpha_n} - 3J_3^{\alpha_n/3} \right] \quad (3.8-23)$$

$$W_V = \sum_{n=1}^N \frac{\mu_n}{\alpha_n} \left[3(J_3^{\alpha_n/3} - 1) + \frac{1}{\beta_n} (J_3^{-\alpha_n \beta_n} - 1) \right] \quad (3.8-24)$$

Notice that $W = W_D + W_V$. This decomposition of the strain

energy density has the advantage that the stresses obtained from the deviatoric and volumetric parts separately are zero when there are no deformations:

$$S_{ij}^D \Big|_{\varepsilon_{ij}=0} = \frac{1}{2} \left(\frac{\partial W_D}{\partial \varepsilon_{ij}} + \frac{\partial W_D}{\partial \varepsilon_{ji}} \right) \Big|_{\varepsilon_{ij}=0} = 0 ,$$

$$S_{ij}^V \Big|_{\varepsilon_{ij}=0} = \frac{1}{2} \left(\frac{\partial W_V}{\partial \varepsilon_{ij}} + \frac{\partial W_V}{\partial \varepsilon_{ji}} \right) \Big|_{\varepsilon_{ij}=0} = 0$$

Notice that W_D contains the volumetric part of the motion through the term $3J_3^{\alpha_n/3}$. Therefore W_D is not entirely deviatoric.

Plane stress analysis: The material is *not* assumed to be totally incompressible. A displacement-based finite element formulation is used.

Plane strain, axisymmetric and 3-D analysis: The material is *not* assumed to be totally incompressible.

Because both W_D and W_V contain the volumetric part of the motion, the mixed u/p formulation cannot be used with the hyper-foam material. A displacement-based formulation is used.

Selection of material constants: The hyper-foam material description used here has 27 constants: $\mu_n, \alpha_n, \beta_n, n=1, \dots, 9$.

The small strain shear modulus and small strain bulk modulus can be written as

$$G = \frac{1}{2} \sum_{n=1}^9 \mu_n \alpha_n \quad (3.8-25)$$

$$\kappa = \sum_{n=1}^9 \left(\beta_n + \frac{1}{3} \right) \mu_n \alpha_n \quad (3.8-26)$$

These moduli must be greater than zero, hence we note that β_n

should be greater than $-1/3$.

When all of the β_n are equal to each other $= \beta$, then the Poisson's ratio is related to β using

$$\beta = \frac{\nu}{1 - 2\nu} \quad (3.8-27)$$

The default values of the material constants are $\mu_1 = 1.85$, $\alpha_1 = 4.5$, $\beta_1 = 9.2$, $\mu_2 = -9.2$, $\alpha_2 = -4.5$, $\beta_2 = 9.2$. It is likely that you will need to change these values in your analysis.

- The hyper-foam material model is generally used for highly compressible elastomers.

If the ratio of the bulk modulus to shear modulus is high (greater than about 10), the material is almost incompressible and we recommend that one of the other hyperelastic materials be used.

3.8.1.5 Sussman-Bathe material model

- The Sussman-Bathe model is based on the following equation:

$$W_D = w(e_1) + w(e_2) + w(e_3) \quad (3.8-28)$$

where $w(e)$ is a function of the principal logarithmic strain (Hencky strain) and e_1 , e_2 , and e_3 are the principal logarithmic strains.

- The primary goal of the model is to fit given uniaxial tension/compression data very well. This goal is accomplished by using a spline to fit the derivative of $w(e)$, as described in detail below.

Of course, when uniaxial tension/compression data is known, a curve fitting approach can, in theory, be used to determine the constants for the other hyperelastic models, e.g. the Ogden material model. But this curve fitting in practice does not provide good fits to the data under many circumstances.

- This strain energy density expression assumes a totally

incompressible material ($I_3 = 1$) and is modified as explained below for plane strain, axisymmetric or 3-D analysis.

- The Sussman-Bathe model is given in the following reference
 ref. T. Sussman and K.J. Bathe, “A model of incompressible isotropic hyperelastic material behavior using spline interpolations of tension-compression test data”, *Commun. Numer. Meth. Engng*, Vol. 25, Issue 1, pp. 53-63, January 2009.
- The following gives a quick summary of the Sussman-Bathe model. In this summary, we assume that the material is totally incompressible. Differences due to slight compressibility are small.

Theoretical background:

- 1) The Cauchy stress τ_i corresponding to the principal strain e_i is

$$\tau_i = \frac{\partial W_D}{\partial e_i} + p = w'(e_i) + p \quad (3.8-29)$$

where $w'(e_i) \equiv dw/de$.

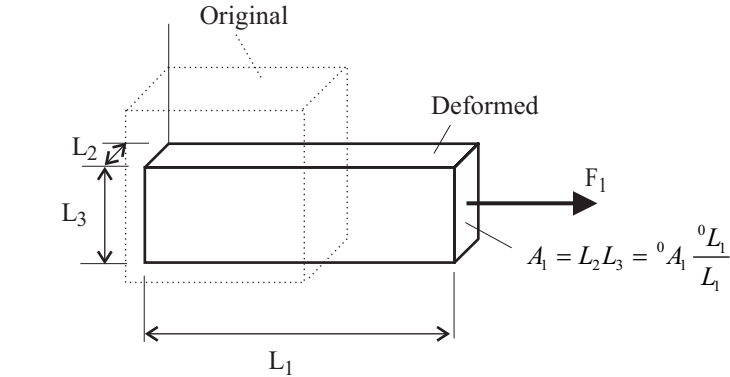
- 2) In uniaxial tension/compression (Figure 3.8-1), $e_1 = e$,
 $e_2 = e_3 = -\frac{1}{2}e$ so

$$\tau = w'(e) - w'(-\frac{1}{2}e) \quad (3.8-30)$$

(3.8-30) can be inverted to obtain

$$w'(e) = \sum_{k=0}^{\infty} \tau \left(\left(-\frac{1}{2}\right)^k e \right) \quad (3.8-31)$$

The series converges when $\tau(e) \rightarrow 0$ as $e \rightarrow 0$.



$$e = e_1 = \ln \frac{L_1}{{}^0 L_1} \quad e_2 = e_3 = -\frac{1}{2} e_1$$

$$\tau_1 = \frac{F_1}{A_1} \quad \tau_2 = \tau_3 = 0$$

Figure 3.8-1: Uniaxial tension/compression test

3) The asymptotic conditions for w are $w'(e) \rightarrow -\infty$ as $e \rightarrow -\infty$; $w'(e) \rightarrow \infty$ as $e \rightarrow \infty$. These asymptotic conditions correspond to the asymptotic conditions of infinite stresses for infinite strains.

4) For a stable material, it is necessary (but not sufficient) that $w''(e) > 0$ for all e . Not all materials for which $\tau'(e) > 0$ have $w''(e) > 0$. For example, the material with

$$\begin{aligned} \tau(e) &= E_T e, \quad e > 0 \\ &= E_C e, \quad e < 0 \end{aligned}$$

where E_T and E_C are constants greater than zero, has $w''(e) > 0$ only if

$$\frac{1}{2} E_T < E_C < 2 E_T$$

5) Given only simple tension data for $\tau(e)$, there are multiple $w'(e)$ that exactly correspond to $\tau(e)$, for positive e only. Two such $w'(e)$ are

$$\begin{aligned} w'(e) &= 0, e < 0 \\ &= \tau(e), e > 0 \end{aligned}$$

and

$$\begin{aligned} w'(e) &= -\tau(-2e), e < 0 \\ &= 0, e > 0 \end{aligned}$$

Hence the material is not uniquely described given only uniaxial tension (or uniaxial compression) data. Both uniaxial tension and uniaxial compression data must be provided to uniquely describe the material.

6) The small-strain Young's modulus E is found by differentiating the uniaxial stress-strain curve, and evaluating at $e = 0$, and, since the material is almost incompressible, the small-strain shear

modulus G is $G = \frac{1}{3}E$. The results are

$$E = \frac{3}{2} w''(0), \quad G = \frac{1}{2} w''(0) \quad (3.8-32a, b)$$

8) The Ogden material model can be considered a special case of (3.8-28), since the Ogden material model can be written in terms of $w'(e)$:

$$w'(e) = \sum_n \mu_n (\exp(\alpha_n e) - 1) \quad (3.8-33)$$

Spline representation of $w'(e)$:

In the Sussman-Bathe model, we choose $w'(e)$ to fit given uniaxial tension/compression data very well, as follows.

The uniaxial tension-compression data is in the form of user-specified data points (e_i, τ_i) . From these data points, we build a non-uniform cubic spline representation of the uniaxial

tension/compression stress-strain data $\tau = \tau(e)$, as shown in Figure 3.8-2. For the non-uniform cubic spline representation of $\tau(e)$,

- 1) A spline segment is placed between two successive user-input data points. The user-input data points need not be equally spaced.
- 2) The range of the cubic spline is between the first and last user-input data points.
- 3) Outside the range of the cubic spline, the slope of $\tau(e)$ is greater than zero. This ensures that the asymptotic conditions of $\tau(-\infty) = -\infty$, $\tau(\infty) = \infty$ are met.

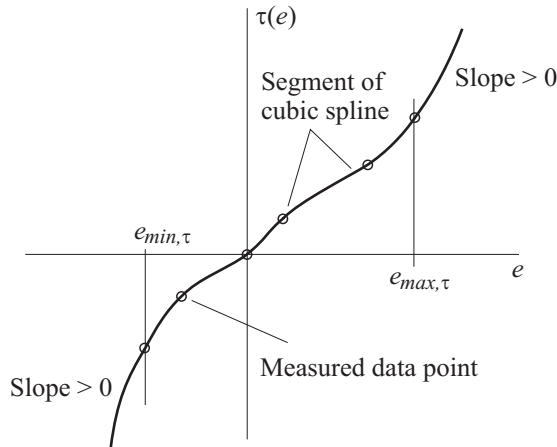


Figure 3.8-2: Uniaxial tension/compression stress-strain spline

Using the non-uniform cubic spline representation of $\tau(e)$ and (3.8-31), we build a uniform cubic spline for $w'(e)$ as shown in Figure 3.8-3. For the uniform cubic spline representation of $w'(e)$,

- 1) The same number of spline segments is used in tension and in compression.
- 2) The range of the cubic spline is the same in tension and in compression. This range includes the range of the user-input data

points.

3) Outside the range of the cubic spline, the slope of $w'(e)$ is greater than zero, whenever possible.

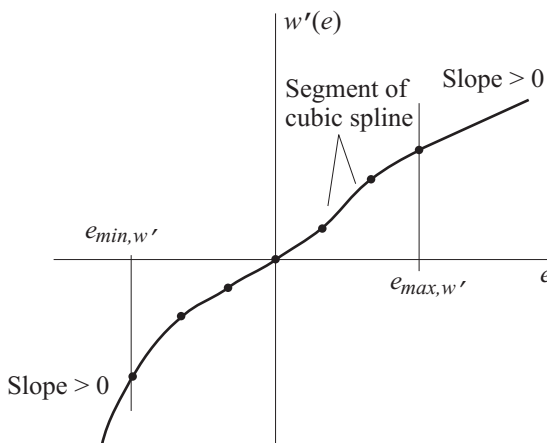


Figure 3.8-3: $w'(e)$ spline

In order to measure the accuracy of the spline representation of $w'(e)$, we define the relative interpolation error

$$r = \max_e \left| \frac{\bar{\tau}(e) - \tilde{\tau}(e)}{\tilde{\tau}(e)} \right| \quad (3.8-34)$$

in which $\bar{\tau}(e)$ is the stress evaluated from the spline representation of $w'(e)$ (using 3.8-30), and $\tilde{\tau}(e)$ is the stress evaluated from the spline representation of $\tau(e)$.

The number of spline segments is automatically chosen to make the interpolation error r smaller than a user-specified value. Typically only a few spline segments need be used for $w'(e)$ in order to reduce the interpolation error to a value smaller than experimental error.

Each cubic spline segment in $w'(e)$ can be written

$$w'(e) = A_{i+1}z + B_{i+1}z^3 + A_i(1-z) + B_i(1-z)^3 \quad (3.8-35)$$

for the segment $e_i \leq e \leq e_{i+1}$ where $z = (e - e_i)/(e_{i+1} - e_i)$. With this definition, $w(e)$ can be written

$$w(e) = C_i + (e_{i+1} - e_i) \left(A_{i+1} \frac{z^2}{2} + B_{i+1} \frac{z^4}{4} + A_i \frac{1 - (1-z)^2}{2} + B_i \frac{1 - (1-z)^4}{4} \right) \quad (3.8-36)$$

The AUI determines the constants A_i , B_i , C_i from uniaxial stress-strain data, as described above. The constants A_i , B_i , C_i are listed in the ADINA output file when the model definition is listed.

Plane stress analysis: The material is assumed to be totally incompressible. Therefore W_V is zero and $W = W_D$. A displacement-based finite element formulation is used, exactly as for the Mooney-Rivlin material model described above.

Plane strain, axisymmetric and 3-D analysis: The material is modeled as compressible (that is, the bulk modulus is not infinite), but you can set the bulk modulus high so that the material is “almost incompressible”.

Equation (3.8-28) is modified by 1) substituting the deviatoric principal strains for the corresponding principal strains, 2) removing the condition $e_1 + e_2 + e_3 = 0$, and 3) adding the volumetric strain energy density

$$W_V = \kappa (J_3 \ln J_3 - (J_3 - 1)) \quad (3.8-37)$$

where κ is the bulk modulus. The relationship between the pressure and the volume ratio is

$$p = -\kappa \ln J_3 = -\kappa (e_1 + e_2 + e_3) \quad (3.8-38)$$

which is a generalization of the small-strain pressure-strain

relationship. Either the u/p formulation (default) or the displacement-based formulation can be used. For comments about the u/p formulation, see the corresponding comments in the Mooney-Rivlin material description.

Data input considerations:

1) Data input is in the form of a set of stress-strain data points, with positive stresses/strains corresponding to uniaxial tension and negative stresses/strains corresponding to uniaxial compression. Compression and tension data are entered together in the same data set.

2) The data set should contain both tension and compression data (compression data is possibly converted from equibiaxial tension data, see below). If the data set contains only tension data, the program will assume that the true stress / true strain curve in compression is a straight line, which is most likely not a good assumption.

3) The stresses and strains in the set of stress-strain data points can be either

- a) True stresses and logarithmic strains
- b) Engineering stresses and engineering strains
- c) Engineering stresses and stretches

In the latter cases, the program internally converts the stresses and strains to true stresses and logarithmic strains.

4) Data points from equibiaxial tension experiments can be converted into equivalent uniaxial compression data. The conversion formulas are:

$$\begin{aligned} e_u &= -2e_b, \quad \lambda_u = \lambda_b^{-2}, \quad {}_0e_u = (1 + {}_0e_b)^{-2} - 1 \\ \tau_u &= -\tau_b, \quad {}_0\sigma_u = -{}_0\sigma_b \lambda_b^3 \end{aligned} \quad (3.8-39)$$

where e_u is the equivalent uniaxial logarithmic strain (< 0), e_b is the equibiaxial logarithmic strain (> 0), λ_u is the equivalent

uniaxial stretch, λ_b is the equibiaxial stretch, ${}_0e_u$ is the equivalent uniaxial engineering strain, ${}_0e_b$ is the equibiaxial engineering strain, τ_u is the equivalent uniaxial true (Cauchy) stress, τ_b is the equibiaxial true (Cauchy) stress, ${}_0\sigma_u$ is the equivalent uniaxial engineering stress, ${}_0\sigma_b$ is the equibiaxial engineering stress. All of these conversion formulas assume that the material is incompressible.

5) The Sussman-Bathe model fits the data so closely that roughness and waviness in the data causes roughness and waviness in the $w'(e)$ splines. The program does not smooth the data in order to eliminate roughness and waviness. If the original data set contains roughness and waviness that should not be present in the analysis, the data set should be smoothed before entering the data into the program.

6) If the data set corresponds to a stable material, then the Sussman-Bathe model is stable, otherwise the Sussman-Bathe model may not be stable.

7) The strain range of the data set should contain the range of strains anticipated during the analysis.

8) Do not confuse uniaxial compression with hydrostatic compression. These two test cases are very different.

3.8.1.6 Eight-chain material model

- The eight-chain model is based on the following expression:

$$W_D = R \left[b\beta + \ln \frac{\beta}{\sinh \beta} - (g\gamma + \ln \frac{\gamma}{\sinh \gamma}) \right] \quad (3.8-40)$$

in which

$$R = \frac{3\mu\lambda_L}{\gamma} \quad (3.8-41)$$

$$b = \frac{\bar{\lambda}}{\lambda_L}, \quad \beta = L^{-1}(b) \quad (3.8-42a, b)$$

$$g = \frac{1}{\lambda_L}, \quad \gamma = L^{-1}(g) \quad (3.8-43a, b)$$

$$\bar{\lambda} = \frac{1}{\sqrt{3}} \sqrt{\lambda_1^2 + \lambda_2^2 + \lambda_3^2} = \frac{1}{\sqrt{3}} \sqrt{I_1} \quad (3.8-44)$$

The function $b = L(\beta)$ is the so-called Langevin function, defined as

$$b = L(\beta) = \coth \beta - (1/\beta) \quad (3.8-45)$$

The inverse Langevin function $\beta = L^{-1}(b)$ cannot be written in closed-form.

- There are two material constants: μ and λ_L . μ is the initial shear modulus and λ_L is the locking stretch. The locking stretch will be discussed in more detail below.

- The eight-chain material model is described in the following references:

ref. E.M. Arruda and M. C. Boyce, “A three-dimensional constitutive model for the large stretch behavior of rubber elastic materials”, *J. Mech. Phys. Solids*, Vol. 41 (2), pp 389-412 (1993).

ref. M. C. Boyce and E.M. Arruda, “Constitutive models of rubber elasticity: a review”, *Rubb. Chem. & Tech.*, Vol. 73 (3), pp 504-523 (2000).

- The Arruda-Boyce material model in Section 3.8.1.3 is an approximation to the eight-chain material model.

Theoretical background

The following gives a brief introduction to the theory.

The material response is governed by the deformations of material chains that connect the corners of a unit cube of material to its center (Fig 3.8-4(a)). This unit cube is selected so that the deformation state is principal, and therefore the deformed side lengths of the unit cube are the principal stretches.

As there are eight chains that connect the corners of the unit cube to its center, the material model is called the "eight-chain model". However, all of these eight chains undergo the same deformation. Therefore attention is focused on the single chain connecting two opposite corners of the unit cube. The initial length of this chain is just ${}^0r = \sqrt{3}$ (Fig 3.8-4(b)) and the current length of the chain is ${}^tr = \sqrt{\lambda_1^2 + \lambda_2^2 + \lambda_3^2}$ (Fig 3.8-4(c)). Thus the stretch of this chain is

$$\lambda_{chain} = \frac{{}^tr}{{}^0r} = \frac{1}{\sqrt{3}} \sqrt{\lambda_1^2 + \lambda_2^2 + \lambda_3^2} \quad (3.8-46)$$

Due to the assumed incompressibility of the material, the stretch λ_{chain} is always greater than or equal to 1.

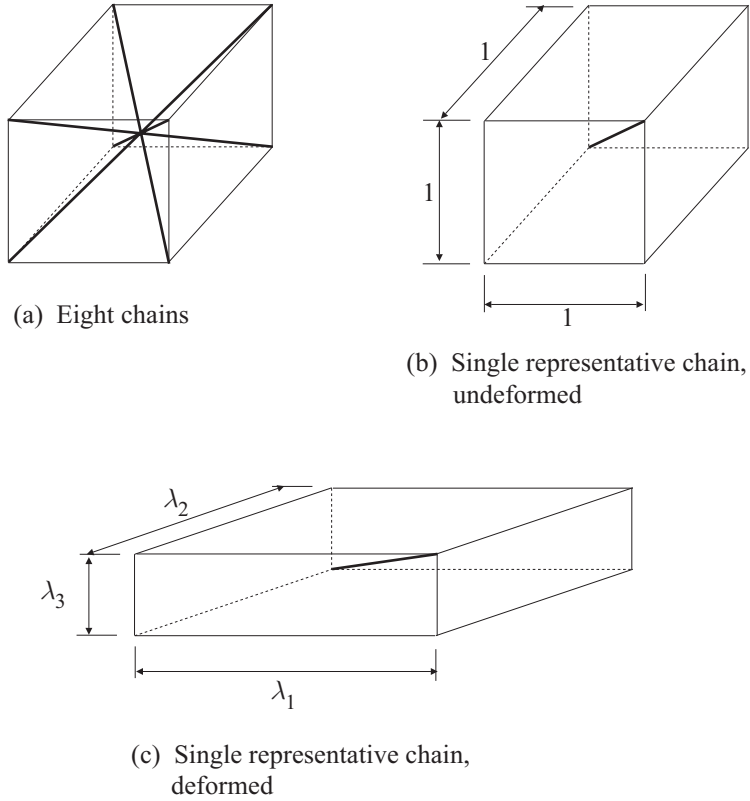


Figure 3.8-4: Unit cube and chains used in eight-chain model

From the paper of Boyce and Arruda (2000), the following expression is given for the strain energy function of the eight-chain model.

$$W = Nk\theta\sqrt{n} \left[\beta\lambda_{chain} + \sqrt{n} \ln \frac{\beta}{\sinh \beta} \right] \quad (3.8-47)$$

Here N is the number of chains in the chain structure, k is Boltzmann's constant, and θ is the absolute temperature. β is defined implicitly by

$$\frac{\lambda_{chain}}{\sqrt{n}} = L(\beta) = \coth \beta - (1/\beta) \quad (3.8-48)$$

For ease of writing, we define $\lambda_L = \sqrt{n}$, $\bar{\lambda} = \lambda_{chain}$, $b = \bar{\lambda} / \lambda_L$ and $R = Nk\theta n$. The result is

$$W = R \left[b\beta + \ln \frac{\beta}{\sinh \beta} \right] \quad (3.8-49)$$

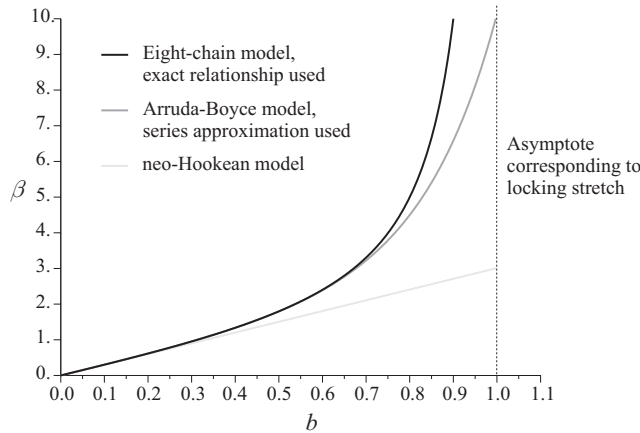
Since this quantity is not zero for zero deformation, we subtract the value of W at zero deformation to obtain the strain energy density expression (3.8-40).

Equation (3.8-41), which gives the value of R in terms of the shear modulus and locking stretch, is obtained by considering the case of simple tension, as described below.

Differentiating equation (3.8-40) with respect to b gives

$$\frac{dW_D}{db} = R\beta \quad (3.8-50)$$

Thus β is seen to be a stress-like quantity conjugate to the strain-like quantity b . A plot of the function $\beta = L^{-1}(b)$ is given in Fig. 3.8-5. Notice that as the stretch $\bar{\lambda}$ approaches the locking stretch, β becomes infinite. Thus the stretch $\bar{\lambda}$ cannot exceed the locking stretch. Also notice that for stretch $\bar{\lambda}$ much smaller than the locking stretch (very small b), the function $\beta = L^{-1}(b)$ is approximated by $\beta \approx 3b$.


 Figure 3.8-5: Relationships between b and β

Also it is evident that the locking stretch λ_L must be greater than 1.

Simple tension

It is instructive to consider the cases of simple tension, pure shear and equibiaxial tension. For simple tension, the engineering stress is calculated using

$${}_0\sigma = \frac{dW_D}{d\lambda} = \frac{dW_D}{db} \frac{db}{d\bar{\lambda}} \frac{d\bar{\lambda}}{d\lambda} = \frac{R\beta}{\lambda_L} \left[\frac{1}{3\bar{\lambda}} (\lambda - \lambda^{-2}) \right] \quad (3.8-51)$$

Here β depends on λ through the relationships $\beta = L^{-1}(b)$,

$$b = \bar{\lambda} / \lambda_L, \quad \bar{\lambda} = \frac{1}{\sqrt{3}} \sqrt{\lambda_1^2 + \lambda_2^2 + \lambda_3^2} = \frac{1}{\sqrt{3}} \sqrt{\lambda^2 + 2\lambda^{-1}}.$$

The derivative of the engineering stress with respect to the stretch, evaluated at zero deformation, is

$$\left. \frac{d{}_0\sigma}{d\lambda} \right|_{\lambda=1} = \left. \frac{d}{d\lambda} \right|_{\lambda=1} \left\{ \frac{R\beta}{\lambda_L} \left[\frac{1}{3\bar{\lambda}} (\lambda - \lambda^{-2}) \right] \right\} = \left. \frac{R\beta}{\lambda_L} \right|_{\lambda=1} = \frac{R\gamma}{\lambda_L} \quad (3.8-52)$$

Since this is equal to the initial Young's modulus $E = 3\mu$, R must

be defined by equation (3.8-41). Notice that, in the limit of large locking stretch,

$$R = \frac{3\mu\lambda_L}{\gamma} \approx \frac{3\mu\lambda_L}{3g} = \mu\lambda_L^2 \quad (3.8-53)$$

and for stretch $\bar{\lambda}$ much smaller than the locking stretch,

$$\frac{R\beta}{\lambda_L} \frac{1}{3\bar{\lambda}} \approx \frac{(\mu\lambda_L^2) \left(3 \frac{\bar{\lambda}}{\lambda_L}\right)}{\lambda_L} \frac{1}{3\bar{\lambda}} = \mu \quad (3.8-54)$$

so the neo-Hookean case is recovered for stretch $\bar{\lambda}$ much smaller than the locking stretch.

The maximum tensile and compressive stretches in simple tension are obtained by setting $\bar{\lambda} = \lambda_L = \frac{1}{\sqrt{3}} \sqrt{\lambda^2 + 2\lambda^{-1}}$. An

approximate solution in tension is obtained by neglecting $2\lambda^{-1}$ and an approximate solution in compression is obtained by neglecting λ^2 . The result is $\frac{2}{3}\lambda_L^{-2} < \lambda < \sqrt{3}\lambda_L$. Thus we see that the

locking stretch λ_L does not correspond directly to an experimentally observed tensile or compressive stretch. However, if by experiment the maximum tensile and compressive stretches are known, the corresponding locking stretch can be obtained with the above expressions.

Pure shear

For pure shear, the engineering stress is calculated using

$${}_0\sigma = \frac{dW_D}{d\lambda} = \frac{R\beta}{\lambda_L} \left[\frac{1}{3\bar{\lambda}} (\lambda - \lambda^{-3}) \right] \quad (3.8-55)$$

Here β depends on λ through the relationships $\beta = L^{-1}(b)$,

$$b = \bar{\lambda} / \lambda_L, \quad \bar{\lambda} = \frac{1}{\sqrt{3}} \sqrt{\lambda_1^2 + \lambda_2^2 + \lambda_3^2} = \frac{1}{\sqrt{3}} \sqrt{\lambda^2 + \lambda^{-2} + 1}.$$

The maximum stretch in pure shear is obtained by setting

$$\bar{\lambda} = \lambda_L = \frac{1}{\sqrt{3}} \sqrt{\lambda^2 + \lambda^{-2} + 1}, \text{ with an approximate maximum}$$

stretch of about $\lambda \approx \sqrt{3}\lambda_L$ (very similar to the maximum tensile stretch in uniaxial tension.)

Equibiaxial tension

For equibiaxial tension, the engineering stress is calculated using

$${}_0\sigma = \frac{1}{2} \frac{dW_D}{d\lambda} = \frac{R\beta}{\lambda_L} \left[\frac{1}{3\bar{\lambda}} (\lambda - \lambda^{-5}) \right] \quad (3.8-56)$$

Here β depends on λ through the relationships $\beta = L^{-1}(b)$,

$$b = \bar{\lambda} / \lambda_L, \quad \bar{\lambda} = \frac{1}{\sqrt{3}} \sqrt{\lambda_1^2 + \lambda_2^2 + \lambda_3^2} = \frac{1}{\sqrt{3}} \sqrt{2\lambda^2 + \lambda^{-4}}.$$

The maximum tensile stretch in equibiaxial tension is obtained by

$$\text{setting } \bar{\lambda} = \lambda_L = \frac{1}{\sqrt{3}} \sqrt{2\lambda^2 + \lambda^{-4}}, \text{ with an approximate maximum}$$

tensile stretch of about $\lambda \approx \sqrt{\frac{3}{2}}\lambda_L$ (significantly different than the maximum tensile stretch in uniaxial tension).

Comparison with Arruda-Boyce model

Although both the eight-chain model and its series approximation are presented in the references by Arruda and Boyce, we use the name Arruda-Boyce to refer only to the series approximation. This is primarily because the series approximation is already widely used under the name Arruda-Boyce.

In terms of terminology given in this section, the Arruda-Boyce

series approximation takes the form

$$\beta \approx 6 \left(\frac{1}{2}b + 6\frac{1}{20}b^3 + 27\frac{11}{1050}b^5 + 108\frac{19}{7000}b^7 + 405\frac{519}{673750}b^9 \right) \quad (3.8-57)$$

in which the original constants of the Arruda-Boyce approximation are shown. This curve is also plotted in Fig. 3.8-5. The Arruda-Boyce series approximation gives results very close to the eight-chain model only when the stretch $\bar{\lambda}$ is not close to the locking stretch. Note that the Arruda-Boyce series approximation does not give a limiting value for the stretch, in other words, the stretch $\bar{\lambda}$ can exceed the locking stretch λ_L when the Arruda-Boyce series approximation is used.

Also, in the Arruda-Boyce model, material constant μ is only an approximation to the initial shear modulus, whereas in the eight-chain model, material constant μ is the initial shear modulus.

For these reasons, the eight-chain model is preferred to the Arruda-Boyce model.

Comparison with neo-Hookean model

The eight-chain model can also be interpreted as an extension of the neo-Hookean model (Mooney-Rivlin model with just C_1 nonzero). In terms of terminology of this section,

$$\beta \approx 3b \quad (3.8-58)$$

This curve is also plotted in Fig. 3.8-5. Material constant C_1 is equal to 1/2 of the initial shear modulus.

Finite element implementation

- The strain energy density expression (3.8-40) assumes a totally

incompressible material ($I_3 = 1$) and is modified as explained below for plane strain, axisymmetric or 3-D analysis.

Plane stress analysis: The material is assumed to be totally incompressible. Therefore W_V is zero and $W = W_D$. A displacement-based finite element formulation is used, exactly as for the Mooney-Rivlin material model described above.

ref. KJB
Section 6.6.2

Plane strain, axisymmetric and 3-D analysis: The material is modeled as compressible (that is, the bulk modulus is not infinite), but you can set the bulk modulus high so that the material is “almost incompressible”.

The eight-chain strain energy density equation is modified by:
1) substituting for the strain invariant I_1 the reduced strain invariant J_1 , 2) removing the condition $I_3 = 1$, and 3) adding the volumetric energy term

$$W_V = \kappa (J_3 \ln J_3 - (J_3 - 1)) \quad (3.8-59)$$

where κ is the small-strain bulk modulus. The relationship between the pressure and the volume ratio is

$$p = -\kappa \ln J_3 \quad (3.8-60)$$

Either the u/p formulation (default) or the displacement-based formulation can be used. For comments about the u/p formulation, see the corresponding comments in the Mooney-Rivlin material description.

When the u/p formulation is used, there should be at least one solution unknown. This is because the constraint equation used in the u/p formulation is nonlinear in the unknown pressures. Therefore equilibrium iterations are required for convergence, even when all of the displacements in the model are prescribed.

Selection of material constants: Both of the material constants μ and λ_L have a clear physical interpretation. μ is the initial shear modulus and λ_L is the locking stretch. The locking stretch

can be estimated by examination of the experimental stress-strain curves for simple tension/compression, pure shear or equibiaxial tension, as explained above.

3.8.1.7 Curve fitting

- You can have the AUI determine the material constants for the Mooney-Rivlin, Ogden, Arruda-Boyce, hyper-foam, or eight-chain materials by a curve-fit to experimental stress-strain data. Note that the Sussman-Bathe model does not require curve-fitting.
- The AUI allows for data from three experimental cases as shown in Figure 3.8-6: simple tension, pure shear and equibiaxial tension. Input from a single experiment from one of these cases or a combination of data from any two, or all three of these experimental cases is allowed.

A single set of constants representing a best fit in the least squares sense is calculated. You must select the chosen order of the approximation (giving the number of constants used in the model). Note: the total number of data points must be greater than the number of material constants to be determined.

For each experimental case, you can enter either the stretch or engineering strain in the strain column and you enter the engineering stress in the stress column. Refer to Figure 3.8-6 to see the definitions of stretch, engineering strain and engineering stress for the experimental cases.

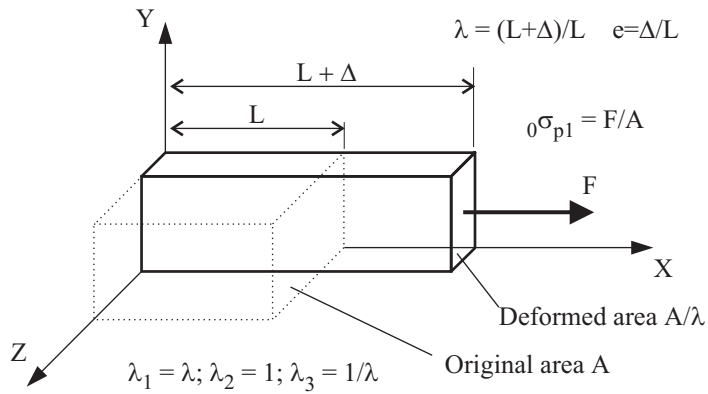
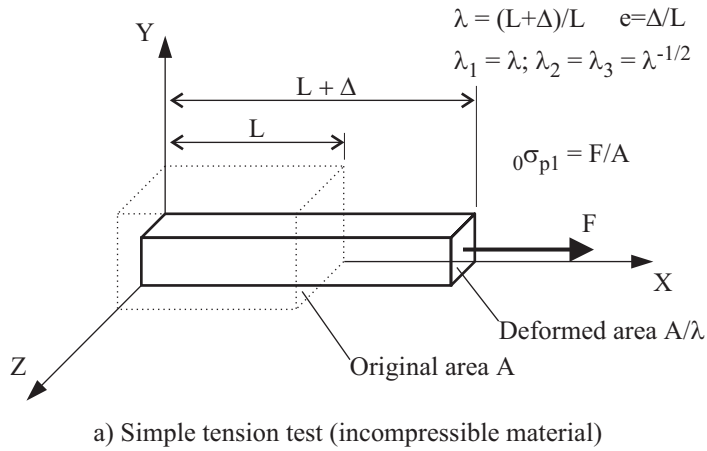
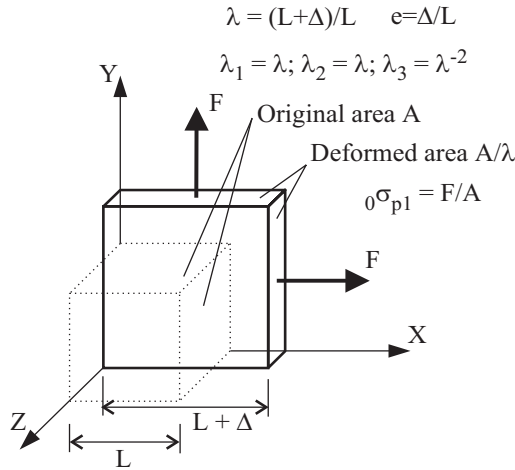


Figure 3.8-6: Experimental test cases for rubber and foam materials



c) Equibiaxial tension test (incompressible material)

Figure 3.8-6: (continued)

Mooney-Rivlin model: The strain energy density for total incompressibility is given in Eq. 3.8-8 with the material constants D_1 and D_2 set to zero.

Given M non-zero material constants, C_i , then M is equal to 2, 5 and 9 for first, second and third order approximations, respectively.

The principal stress in the direction of principal stretch λ_i is given by

$${}_0\sigma_{pi} = \frac{\partial W}{\partial \lambda} = {}_0\sigma_{pi}(C_i, \lambda) \quad (i=1, \dots, M) \quad (3.8-61)$$

Notice that this is the engineering principal stress, that is, the force divided by the original area of the specimen. Thus, we have an expression for the principal stress that is linear in terms of the material constants C_i , and dependent on the principal stretch determined from the allowed experimental cases. Hence, given a set of data points (λ_k, σ_k) from one or more of the test cases, we can establish the least squares fit by a curve of the form given by Eq. (3.8-61).

Consider the sum of discrete error terms at the given data points of one of the experimental test cases (simple tension, pure shear or equibiaxial tension)

$$S = \sum_{k=1}^N \left(\sigma_k - \hat{\sigma}_{pl}(\lambda_k) \right)^2 \quad (3.8-62)$$

where $\hat{\sigma}_{pl}$ is the approximating function dependent on the constants C_i ($i = 1, \dots, M$) given by Eq. (3.8-61). We can expand this sum to include data from all three test cases, i.e.:

$$S = \sum_{j=1}^3 \left(\sum_{k=1}^{N^j} \left(\sigma_k^j - \hat{\sigma}_1^j(\lambda_k^j) \right)^2 \right) \quad (3.8-63)$$

where the superscript j denotes the experimental test case (simple tension $j = 1$, pure shear $j = 2$, equibiaxial tension $j = 3$). Note: the form of $\hat{\sigma}_1^j$ differs for $j = 1, 2, 3$ due to the different principal stretch terms λ_2 and λ_3 .

The least squares fit is determined by minimizing S in Eq. (3.8-63) with respect to the constants C_i , which gives M algebraic equations for the constants C_i ($i = 1, \dots, M$).

Ogden model: The strain energy density function is, for total incompressibility, given by Eq. (3.8-15). Thus, for an M -term Ogden model, there are $2M$ constants, i.e., the pairs (μ_i, α_i) for $i = 1, \dots, M$.

Assuming the principal stretches for the three test cases shown in Figure 3.8-4, the following form for the principal stress ${}_0\sigma_{pl}$ can be derived:

$${}_0\sigma_{pl} = \sum_{i=1}^M \mu_i \left(\lambda^{\alpha_i-1} - \lambda^{c\alpha_i-1} \right) \quad (3.8-64)$$

with $c = -0.5$ for simple tension, -1.0 for pure shear, -2.0 for equibiaxial tension.

Equation (3.8-64) is linear in terms of the coefficients μ_i , but nonlinear in terms of the exponents α_i . A nonlinear least squares algorithm could be employed to iterate to a solution for the

constants (μ_i, α_i) . However, in our curve-fitting implementation in the AUI, you specify the desired exponent terms α_i (with the default $\alpha_i = i$), and the least squares fit of the resulting function yields the corresponding coefficients μ_i . You can select the number of terms M desired ($1 \leq M \leq 9$).

Arruda-Boyce model: The strain energy density function is, for total incompressibility, given by Eq. (3.8-19). There are only two material constants, μ and λ_L .

Assuming the principal stretches for the three test cases shown in Figure 3.8-6, the following forms for the principal stress ${}_0\sigma_{pl}$ can be derived:

Uniaxial tension:

$${}_0\sigma_{pl} = 2\mu(\lambda - \lambda^{-2}) \sum_{i=1}^5 \frac{iC_i}{\lambda_L^{2i-2}} I_1^{i-1}, \quad I_1 = \lambda^2 + 2\lambda^{-1} \quad (3.8-65)$$

Pure shear:

$${}_0\sigma_{pl} = 2\mu(\lambda - \lambda^{-3}) \sum_{i=1}^5 \frac{iC_i}{\lambda_L^{2i-2}} I_1^{i-1}, \quad I_1 = \lambda^2 + \lambda^{-2} + 1 \quad (3.8-66)$$

Equibiaxial tension:

$${}_0\sigma_{pl} = 2\mu(\lambda - \lambda^{-5}) \sum_{i=1}^5 \frac{iC_i}{\lambda_L^{2i-2}} I_1^{i-1}, \quad I_1 = 2\lambda^2 + \lambda^{-4} \quad (3.8-67)$$

The AUI uses a nonlinear curve-fitting algorithm to determine μ and λ_L .

Hyper-foam model: The strain energy density function is given by Eq. (3.8-22). However, we seek a curve-fit for which all of the β_n terms are equal. Thus, for an M-term hyper-foam model, there are $2M + 1$ material constants, i.e., the constants (μ_n, α_n) for $n = 1, \dots, M$, and β .

First β is determined using information about the lateral stretch λ_T . There are two options for entering the lateral stretch

information:

1) Specify the Poisson's ratio. β is then calculated using (3.8-27).

2) Enter the lateral stretches as part of the stress-strain data input. Then β is determined with the use of the following formulas:

Uniaxial tension:

$$\lambda_T = \lambda^{-\frac{\beta}{1+2\beta}} \quad (3.8-68)$$

Pure shear:

$$\lambda_T = \lambda^{-\frac{\beta}{1+\beta}} \quad (3.8-69)$$

Equibiaxial tension:

$$\lambda_T = \lambda^{-\frac{2\beta}{1+\beta}} \quad (3.8-70)$$

Since each of these formulas can be also written in the form $\ln \lambda_T = -f(\beta) \ln \lambda$, a linear curve-fit is performed to obtain β .

Next, the constants (μ_n, α_n) are determined. Assuming the principal stretches for the three test cases shown in Figure 3.8-6, the following form for the principal stress ${}_0\sigma_{pI}$ can be derived:

$${}_0\sigma_{pI} = \frac{1}{\lambda} \sum_{n=1}^M \mu_n \left(\lambda^{\alpha_n} - J_3^{-\alpha_n \beta} \right) \quad (3.8-71)$$

where $J_3 = \lambda \lambda_T^2$ for uniaxial tension, $J_3 = \lambda \lambda_T$ for pure shear and $J_3 = \lambda^2 \lambda_T$ for equibiaxial tension. (3.8-71) is then used as the basis of a nonlinear curve-fitting algorithm to obtain the constants (μ_n, α_n) .

Eight-chain model: The strain energy density function is, for total incompressibility, given by Eq. (3.8-40). There are only two material constants, μ and λ_L .

Assuming the principal stretches for the three test cases shown in Figure 3.8-6, the principal stress ${}_0\sigma_{pI}$ is given by equations (3.8-51), (3.8-55), (3.8-56).

The AUI uses a nonlinear curve-fitting algorithm to determine μ and λ_L .

Notes for curve-fitting for the Mooney-Rivlin and Ogden material models: The system of equations used to solve for the least squares fitted constants is likely to be ill-conditioned. Direct use of Gaussian elimination may yield constants with large magnitudes, which are balanced to cancel out when the fitted function $\hat{\sigma}$ is evaluated. Furthermore, and perhaps more importantly, there is no guarantee that the fitted function has physically feasible behavior outside the range of the data supplied.

The reason for the ill-conditioning of the least squares system matrix is that the data may be fitted equally well by two or more of the "basis" functions or combinations thereof, thus rendering the matrix nearly singular.

The AUI provides an alternative matrix solution algorithm to the Gauss-Jordan elimination technique, namely that of Singular Value Decomposition (SVD) described in

ref. Forsythe, G.E., Malcom, M.A. and Moler, C.B.,
Computer Methods for Mathematical Computations,
Chapter 9, Prentice Hall, Inc. Englewood Cliffs, NJ,
1977.

The essence of the SVD algorithm is that combinations of the basis functions which correspond to near zero eigenvalues of the matrix system are eliminated.

Successive elimination of the near singular terms always increases the error (in the least squares sense). However, the increase in error is often small compared to the overall improvement in the fitted solution – the constants obtained are

usually smaller in magnitude and physically reasonable.

The SVD algorithm is controlled by successively eliminating combinations corresponding to the smallest eigenvalues until a fitted function is obtained which gives a monotonic increase in stress with strain.

Depending on data, the number of terms eliminated from a higher order approximation may be large, giving a poor fit, even though the monotonicity criterion has been achieved.

As an example, consider the curve fitting results illustrated in Fig. 3.8-7. Fig. 3.8-7(a) shows the fitted curves for a 9-term ($\alpha_i = i, i = 1, \dots, 9$) Ogden model approximating a set of data points for a simple tension test. The sum of error squares S is given in the key of the graphs.

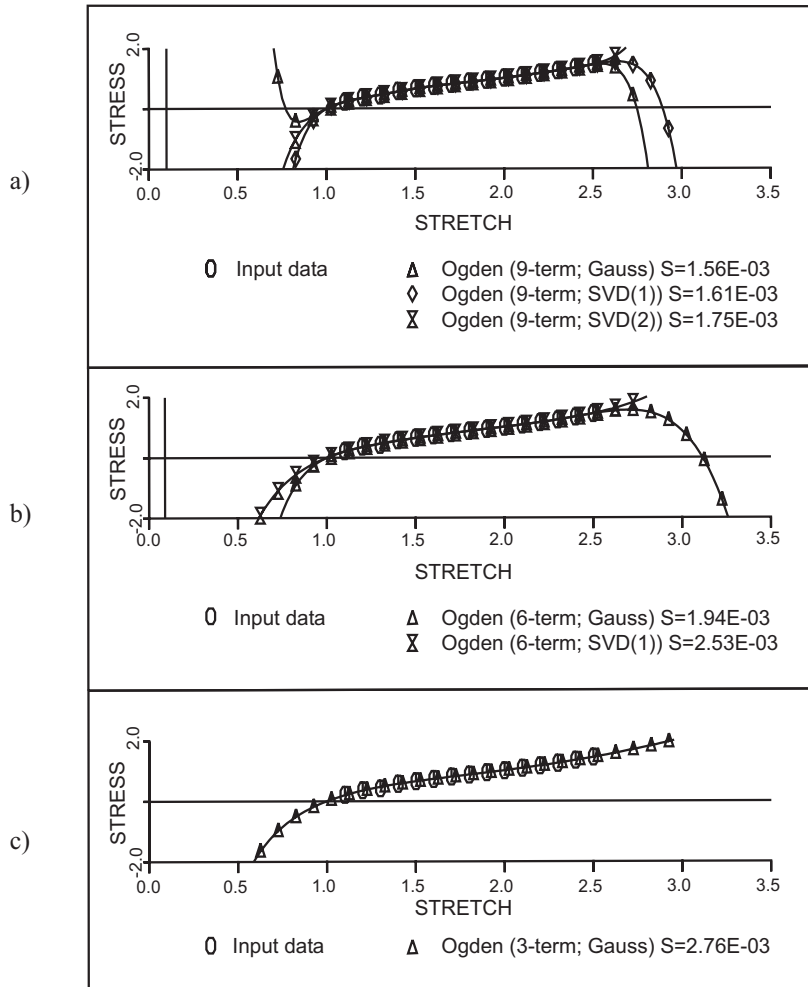


Figure 3.8-7: Curve fitting results for Ogden material model

While the Gaussian elimination curve has the minimum S , its solution is asymptotically unacceptable. Two terms were removed to give a monotonic increasing solution (SVD(2)) which still gives a good approximation to the input data. Fig. 3.8-7(b) shows the fitted curves for a 6-term Ogden model for the same input data. A monotonic increasing solution was obtained after removal of one term (SVD(1)), while the Gauss elimination solution was not asymptotically feasible. For the 3-term Ogden fit to the data, the Gauss elimination solution was accepted since it provided a

monotonic increasing curve, as shown in Fig. 3.8-7(c).

Notes for curve-fitting for the Arruda-Boyce and hyper-foam material models: The Marquardt-Levenberg algorithm from the following reference is used in the curve fitting:

ref. W.H. Press, B.P. Flannery, S.A. Teukolsky and W.T. Vetterling, *Numerical Recipes – the Art of Scientific Computing (Fortran Version)*, Cambridge University Press, pp 521-528 (1990).

For the curve-fitting procedure for the hyper-foam material, the starting values for the material constants are the user-provided values for the material constants. It is suggested that the curve fitting first be done using one term only. If the curve-fit is not accurate enough, then a two-term curve fit can be done with starting values for the first term obtained from the curve-fit for the first term. This procedure can be repeated for each additional term until an accurate curve fit is obtained.

General comments: It can be seen that the curve fitting technique provided gives a valuable tool, but one which must be used with caution. Always check the resulting fitted function. If a physically reasonable fit is not achieved, we recommend the following strategy

- a) Use a lower order approximation.
- b) Use a different material model
- c) Provide more data points, covering a greater range of anticipated strains, including values for compression ($\lambda < 1$).

Specification of input: All of the information required for the curve fitting is contained in a data set of type curve-fitting. The curve-fitting data contains the numbers of the data sets for each of the three test cases, the order of the curve fit and other information required for the curve fit.

3.8.2 Viscoelastic effects

- Viscoelastic effects can be included in the Mooney-Rivlin,

Ogden, Arruda-Boyce, hyper-foam, Sussman-Bathe, and eight-chain material models.

Two viscoelastic models are available:

- ▶ Holzapfel viscoelasticity (Section 3.8.2.1)
- ▶ Bergström-Boyce viscoelasticity (Section 3.8.2.2)

3.8.2.1 Holzapfel viscoelasticity

In the following, we give a brief discussion of the Holzapfel model for finite strain viscoelasticity, see the following references for more information:

- ref. G. A. Holzapfel, “On large strain viscoelasticity: continuum formulation and finite element applications to elastomeric structures”, *Int. J. Num. Meth. Engng.*, Vol. 39, pp 3903-3926, 1996.
- ref. G. A. Holzapfel, *Nonlinear solid mechanics. A continuum approach for engineering*. John Wiley & Sons, Chichester, pp 278-295, 2000.
- ref. G. A. Holzapfel, “Biomechanics of soft tissue”, in Lemaitre (ed.), *Handbook of Materials Behavior Models: Nonlinear Models and Properties*, Academic Press, 2001, pp 1057-1071.

Equivalent 1D model: The equivalent 1D model is shown in Fig 3.8-8. It is the same as a generalized Maxwell model with many chains. A generic chain is denoted with superscript α , as shown in the figure.

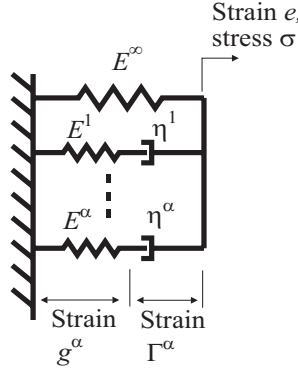


Figure 3.8-8: Equivalent 1D model for viscoelastic effects

The spring E^∞ is equivalent to the elastic stiffness of the model. Each chain contains a spring with stiffness E^α and dashpot with viscosity η^α . (Note that the superscripts ∞ and α do not denote exponentiation.) The strain in each chain is the sum of the strain in the spring g^α and the strain in the dashpot Γ^α . The observed stress is

$$\sigma = \sigma^\infty + \sum_{\alpha} q^\alpha \quad (3.8-72)$$

where $\sigma^\infty = E^\infty e$ is the elastic stress and $q^\alpha = E^\alpha g^\alpha = \eta^\alpha \dot{\Gamma}^\alpha$ is the stress in chain α . Using the definition $\tau^\alpha = \frac{\eta^\alpha}{E^\alpha}$ and the assumption $E^\alpha = \beta^\alpha E^\infty$, the following expression is obtained:

$$\dot{q}^\alpha + \frac{1}{\tau^\alpha} q^\alpha = \beta^\alpha \dot{\sigma}^\infty \quad (3.8-73)$$

Assuming that ${}^0 q^\alpha = 0$, (3.8-73) can be written in convolution form as

$$q^\alpha = \int_0^t \exp\left(-\frac{t-t'}{\tau^\alpha}\right) \beta^\alpha {}^{t'} \dot{\sigma}^\infty dt' \quad (3.8-74)$$

from which the total stress is

$$\sigma = \int_0^t E^\infty \left[1 + \sum_\alpha \beta^\alpha \exp\left(-\frac{t-t'}{\tau^\alpha}\right) \right] \dot{e} dt' \quad (3.8-75)$$

Evidently the relaxation modulus is

$E(t) = E^\infty \left[1 + \sum_\alpha \beta^\alpha \exp\left(-\frac{t}{\tau^\alpha}\right) \right]$ which is a Prony series expression.

The dissipation in dashpot α is

$$D^\alpha = q^\alpha \dot{\Gamma}^\alpha = q^\alpha (\dot{e} - \dot{g}^\alpha) = q^\alpha \left(\dot{e} - \frac{1}{\beta^\alpha} \frac{\dot{q}^\alpha}{E^\infty} \right) \quad (3.8-76)$$

and the total dissipation is $D = \sum_\alpha D^\alpha$. In the above, the

viscoelastic material constants for each chain are τ^α and β^α .

Potential-based 1D model: The 1D model can be written in terms of a potential as follows:

$$\Psi = \Psi^\infty(e) + \sum_\alpha \Psi^\alpha(g^\alpha) \quad (3.8-77)$$

where $\Psi^\infty(e) = \frac{1}{2} E^\infty e^2$ is the strain energy of the elastic chain

and $\Psi^\alpha(g^\alpha) = \frac{1}{2} E^\alpha (g^\alpha)^2$ is the strain energy in the spring of

chain α . In terms of β^α , $\Psi^\alpha(g^\alpha) = \beta^\alpha \Psi^\infty(g^\alpha)$. The 1D model is recovered by defining

$$\sigma = \frac{\partial \Psi}{\partial e} \bigg|_{\Gamma^\alpha \text{ fixed}}, \quad q^\alpha = - \frac{\partial \Psi}{\partial \Gamma^\alpha} \bigg|_{e \text{ fixed}} \quad (3.8-78a,b)$$

Notice that (3.8-72) and (3.8-78a) imply

$$q^\alpha = \frac{\partial \Psi^\alpha}{\partial e} \bigg|_{\Gamma^\alpha \text{ fixed}} = \frac{\partial \Psi^\alpha}{\partial g^\alpha}.$$

Finite strain model: The finite strain model is derived from the potential-based 1D model as follows. The elastic potential is defined as

$$\Psi^\infty(\boldsymbol{\varepsilon}) = W(\boldsymbol{\varepsilon}) \quad (3.8-79)$$

where $W(\boldsymbol{\varepsilon}_{ij})$ is the strain energy density from the elastic part of the material model. The potential of each chain α is defined as

$$\begin{aligned} \Psi^\alpha(\boldsymbol{\varepsilon}_{ij}, \Gamma_{ij}^\alpha) &= \beta^\alpha W(G_{ij}^\alpha), \text{ usage=combined} \\ &= \beta^\alpha W_D(G_{ij}^\alpha), \text{ usage=deviatoric} \\ &= \beta^\alpha W_V(G_{ij}^\alpha), \text{ usage=volumetric} \end{aligned} \quad (3.8-80a,b,c)$$

in which the usage flag (which is a user-input flag) determines whether the entire elastic strain energy density, deviatoric strain energy density or volumetric strain energy density is taken for chain α . Here G_{ij}^α is analogous to the strain in the 1D spring g^α , and we assume $G_{ij}^\alpha = \boldsymbol{\varepsilon}_{ij} - \Gamma_{ij}^\alpha$. Note that with this definition of G_{ij}^α , we have

$$S_{ij} = \frac{\partial \Psi}{\partial \boldsymbol{\varepsilon}_{ij}} \bigg|_{\Gamma_{ij}^\alpha \text{ fixed}}, \quad Q_{ij}^\alpha = \frac{\partial \Psi^\alpha}{\partial \boldsymbol{\varepsilon}_{ij}} \bigg|_{\Gamma_{ij}^\alpha \text{ fixed}} = - \frac{\partial \Psi}{\partial \Gamma_{ij}^\alpha} \bigg|_{\boldsymbol{\varepsilon}_{ij} \text{ fixed}} = \frac{\partial \Psi}{\partial G_{ij}^\alpha} \quad (3.8-81a,b)$$

where S_{ij} are the 2nd Piola-Kirchhoff stresses and Q_{ij}^α is analogous to the stress q^α .

Following exactly the same arguments as in the 1D case, we obtain

$$\dot{Q}_{ij}^{\alpha} + \frac{1}{\tau^{\alpha}} Q_{ij}^{\alpha} = \beta^{\alpha} \dot{S}_{ij}^{\infty} \quad (3.8-82)$$

Assuming that ${}^0Q_{ij}^{\alpha} = 0$, (3.8-82) can be written in convolution form as

$$Q_{ij}^{\alpha} = \int_0^t \exp\left(-\frac{t-t'}{\tau^{\alpha}}\right) \beta^{\alpha} {}^tS_{ij}^{\infty} dt' \quad (3.8-83)$$

and (3.8-83) can be numerically approximated using

$${}^{t+\Delta t}Q_{ij}^{\alpha} = \exp\left(-\frac{\Delta t}{\tau^{\alpha}}\right) {}^tQ_{ij}^{\alpha} + \beta^{\alpha} \frac{1 - \exp\left(-\frac{\Delta t}{\tau^{\alpha}}\right)}{\frac{\Delta t}{\tau^{\alpha}}} \left({}^{t+\Delta t}S_{ij}^{\infty} - {}^tS_{ij}^{\infty}\right). \quad (3.8-84)$$

(3.8-84) is exact when S_{ij}^{α} does not change during the time step, and is more accurate than the formula given by Holzapfel:

$${}^{t+\Delta t}Q_{ij}^{\alpha} = \exp\left(-\frac{\Delta t}{\tau^{\alpha}}\right) {}^tQ_{ij}^{\alpha} + \beta^{\alpha} \exp\left(-\frac{\Delta t}{2\tau^{\alpha}}\right) \left({}^{t+\Delta t}S_{ij}^{\infty} - {}^tS_{ij}^{\infty}\right) \quad (3.8-85)$$

especially when $\frac{\Delta t}{\tau^{\alpha}}$ is large.

Dissipation calculations: If the dissipation is required, it is calculated using

$$D^{\alpha} = Q_{ij}^{\alpha} \dot{\Gamma}_{ij}^{\alpha} = Q_{ij}^{\alpha} \left(\dot{\varepsilon}_{ij} - \dot{G}_{ij}^{\alpha}\right) \quad (3.8-86)$$

where

$$\frac{\partial^2 \Psi^{\alpha}}{\partial G_{ij}^{\alpha} \partial G_{rs}^{\alpha}} \dot{G}_{rs}^{\alpha} = \dot{Q}_{ij}^{\alpha} \quad (3.8-87)$$

is used to compute the unknown \dot{G}_{rs}^α from the known \dot{Q}_{ij}^α .

If usage=combined,

$$\frac{\partial^2 \Psi^\alpha}{\partial G_{ij}^\alpha \partial G_{rs}^\alpha} = \beta^\alpha \frac{\partial^2 W}{\partial \varepsilon_{ij} \partial \varepsilon_{rs}} = \beta^\alpha C_{ijrs} . \quad (3.8-88)$$

where the tensor C_{ijrs} is evaluated at the strain state G_{ij}^α . The dissipation calculation requires the solution of a set of simultaneous linear equations of at most order 6 (in the 3D case) at each integration point.

If usage=deviatoric,

$$\frac{\partial^2 \Psi^\alpha}{\partial G_{ij}^\alpha \partial G_{rs}^\alpha} = \beta^\alpha \frac{\partial^2 W_D}{\partial \varepsilon_{ij} \partial \varepsilon_{rs}} = \beta^\alpha (C_D)_{ijrs} \quad (3.8-89)$$

where the tensor $(C_D)_{ijrs}$ is evaluated at the strain state G_{ij}^α . Here, the dissipation calculation requires a singular value decomposition of $(C_D)_{ijrs}$, since $(C_D)_{ijrs}$ has a zero eigenvalue. A similar situation applies when usage=volumetric, except that the corresponding material tensor has only one nonzero eigenvalue.

The procedure given in (3.8-86) to (3.8-89) is only approximate, since the fundamental assumption $G_{ij}^\alpha = \varepsilon_{ij} - \Gamma_{ij}^\alpha$ strictly speaking only holds for small strain analysis.

Restrictions and recommendations: The allowed values of the usage flag depend upon the material model and finite element type, as shown in Table 3.8-2.

In view of the restrictions, we recommend that usage=“combined” be used in conjunction with the hyper-foam material model, and that usage=“deviatoric” be used in conjunction with the Mooney-Rivlin, Ogden, Arruda-Boyce, Sussman-Bathe and eight-chain material models.

Time-temperature superposition: The preceding derivation

assumes that the viscoelastic response is not temperature-dependent. One method of including the effects of temperature is the method of time-temperature superposition.

In time-temperature superposition, the actual time t is replaced by the reduced time ζ . The relationship between the actual time and reduced time is given by

$$\frac{d\zeta}{dt} = \frac{1}{a_T(' \theta)} \quad (3.8-90)$$

where $' \theta$ is the temperature and $a_T(' \theta)$ is the shift function.

Table 3.8-2: Allowed values of the usage flag

	Mooney-Rivlin, Ogden, Arruda-Boyce, Sussman-Bathe, eight- chain		hyper-foam	
	Plane stress ¹	Plane strain, axisymmetric, 3D ²	Plane stress ³	Plane strain, axisymmetric, 3D
usage=combined	yes	no	yes	yes
usage=deviatoric	yes	yes	no	yes
usage=volumetric	no	no	no	yes

1) Usage cannot be equal to “volumetric”. This is because the material is assumed to be fully incompressible, hence the volumetric strain energy density is zero.

2) When the u/p formulation is used, the usage flag cannot be “combined” or “volumetric”. This is because the modification to the volumetric stresses caused when the usage flag is “combined” or “volumetric” is not taken into account in the u/p formulation.

3) The only allowable value of the usage flag is “combined”. This is because the out-of-plane stress component S_{xx} must be zero, and in the Holzapfel finite strain viscoelastic model, the only way that this can happen is if S_{xx}^{∞} is zero.

Evidently

$${}^t\zeta = \int_0^t \frac{1}{a_T({}^t\theta)} dt' \quad (3.8-91)$$

The shift function used here is either the WLF shift function,

$$\log_{10} a_T({}^t\theta) = -\frac{C_1({}^t\theta - \theta_{ref})}{C_2 + {}^t\theta - \theta_{ref}} \quad (3.8-92)$$

or the Arrhenius shift function

$$\begin{aligned} \log_{10} a_T({}^t\theta) &= C_1 \left(\frac{1}{{}^t\theta} - \frac{1}{\theta_{ref}} \right), \quad {}^t\theta \geq \theta_{ref} \\ &= C_2 \left(\frac{1}{{}^t\theta} - \frac{1}{\theta_{ref}} \right), \quad {}^t\theta < \theta_{ref} \end{aligned} \quad (3.8-93)$$

where θ_{ref} is the reference temperature and C_1 , C_2 are material constants. Notice that as ${}^t\theta$ increases, $a_T({}^t\theta)$ decreases and $\frac{d\zeta}{dt}$ increases. When using the Arrhenius shift function, the temperature unit must be absolute (either Kelvin or Rankine).

For the viscoelastic model used here, the differential equation of the 1D model (3.8-73) becomes

$$\frac{dq^\alpha}{d\zeta} + \frac{1}{\tau^\alpha} q^\alpha = \beta^\alpha \frac{d\sigma^\infty}{d\zeta} \quad (3.8-94)$$

and using (3.8-90), (3.8-94) can be written as

$$\dot{q}^\alpha + \frac{1}{a_T(\theta)\tau^\alpha} q^\alpha = \beta^\alpha \dot{\sigma}^\infty \quad (3.8-95)$$

It is seen that the effect of temperature is to modify the time constants. As the temperature increases, the modified time constants become smaller, that is, the material relaxes more quickly.

The convolution equation of the finite strain model becomes

$$Q_{ij}^{\alpha} = \int_0^{t\zeta} \exp\left(-\frac{\zeta - \zeta'}{\tau^{\alpha}}\right) \beta^{\alpha} \frac{d^{t'} S_{ij}^{\infty}}{d\zeta} d\zeta' \quad (3.8-96)$$

and (3.8-96) is numerically approximated by

$${}^{t+\Delta t} Q_{ij}^{\alpha} = \exp\left(-\frac{\Delta\zeta}{\tau^{\alpha}}\right) {}^t Q_{ij}^{\alpha} + \beta^{\alpha} \frac{1 - \exp\left(-\frac{\Delta\zeta}{\tau^{\alpha}}\right)}{\frac{\Delta\zeta}{\tau^{\alpha}}} \left({}^{t+\Delta t} S_{ij}^{\infty} - {}^t S_{ij}^{\infty}\right) \quad (3.8-97)$$

The only additional consideration is to calculate $\Delta\zeta$, and this is done using

$$\Delta\zeta = \int_t^{t+\Delta t} \frac{1}{a_T({}^{t'}\theta)} dt' \quad (3.8-98)$$

This integration is performed numerically assuming that $\ln a_T(\theta)$ varies linearly over the time step.

Heat generation: A user-specified fraction of the energy dissipated by the viscoelastic model can be considered as heat generation. This heat generation can cause heating in a TMC (thermo-mechanical-coupling) analysis.

Specification of input: To add viscoelastic effects to the rubber-like material model, you need to define the viscoelastic data using a data set of type rubber-viscoelastic, then specify the number of the rubber-viscoelastic data set in the rubber-like material model.

The rubber-viscoelastic data set includes:

- ▶ A flag indicating whether the WLF or Arrhenius shift function is used.
- ▶ The shift function material constants C_1 , C_2
- ▶ A table with one row for each chain. Each row contains β^α , τ^α , the heat generation factor (fraction of dissipation considered as heat generation, default value is 0.0), and the usage flag (default value is deviatoric). There is no restriction on the number of chains permitted. The usage flag can be different for each chain.

In order to include time-temperature superposition, the material must have a TRS temperature dependence.

- It is seen that the dissipation calculation can be quite expensive. Furthermore the dissipation is not required for the stress solution. Therefore it is the default in ADINA to not perform the dissipation calculation. The dissipation is only calculated for the chain α when the heat generation factor is non-zero.

3.8.2.2 Bergström-Boyce viscoelasticity

- This subsection describes ADINA's implementation, with some modifications, of the model described by:

ref. J.S. Bergström and M.C. Boyce, "Constitutive modeling of the large strain time-dependent behavior of elastomers," *J. Mech. Phys. Solids*, Vol. 46, pp 931–954, 1998.

The viscoelastic model of Bergström and Boyce (1998) captures well the time-dependent effects of macro-molecular 'reptation,' or the gradual, relative movement of entangled polymer chains.

This viscoelastic model is especially well suited for modeling the finite viscoelastic response of filled rubbers, nitrile rubbers, silicone rubbers, and other elastomers. It is available to users via

the command RUBBER-VISCOELASTIC BERGSTROM-BOYCE.

- The Bergström-Boyce viscoelastic model is available for the following elements types: 3D solid, 2D solid axisymmetric and plane strain, and 3D shell.

3.8.2.2.1 Rheological representation

This viscoelastic model is comprised of 2 parallel chains and is implemented within ADINA's Parallel-Network Framework (see Section 3.17). The first chain A is purely hyperelastic, and the second chain B includes both viscous (time-dependent) and elastic effects.

The parallel network assumption requires $\mathbf{X} = \mathbf{X}_A = \mathbf{X}_B$, and the multiplicative Kröner-Lee decomposition for viscoelastic network B is

$$\mathbf{X}_B = \mathbf{X}_B^e \mathbf{X}_B^v, \quad (3.8-99)$$

where \mathbf{X}_B^e and \mathbf{X}_B^v are the elastic and inelastic distortions, respectively.

3.8.2.2.2 Elastic response

Bergström and Boyce (1998) employ the eight-chain hyperelastic function for both the purely hyperelastic chain A and the elastic portion of chain B . ADINA, however, permits users to employ Bergström-Boyce viscoelastic effects with any of ADINA's hyperelastic models.

Users may scale the effects of the hyperelastic response in network B using the factor `RATIOBA`. For example, if `RATIOBA` = 1.6 and if both elastic networks are experiencing the same motions, then the hyperelastic contribution of network B will be 60% greater than that of network A .

When used with the Arruda-Boyce or eight-chain hyperelastic models, `RATIOBA` is equivalent to the ratio $C_R^{(B)} / C_R^{(A)}$ as

employed by Bergström and Boyce.

3.8.2.2.3 Viscous response

Bergström and Boyce (1998) postulate time-dependent viscous behavior in network B of the form

$$\dot{\gamma}_B = K_1 \left(\lambda_B^v - 1 \right)^{K_2} \left(\tau_B \right)^{K_3}, \quad (3.8-100)$$

where $\lambda_B^v = \sqrt{\frac{\text{tr } \mathbf{C}_B^v}{3}}$ is the inelastic chain stretch, $\mathbf{C}_B^v = \mathbf{X}_B^v \mathbf{X}_B^v$ is the inelastic right Cauchy-Green tensor, and

$$\tau_B = \sqrt{\frac{1}{2} \boldsymbol{\tau}_{B_{\text{dev}}} : \boldsymbol{\tau}_{B_{\text{dev}}}}, \quad (3.8-101)$$

where $\boldsymbol{\tau}_{B_{\text{dev}}} = \boldsymbol{\tau}_B - \frac{1}{3} \text{tr}(\boldsymbol{\tau}_B) \mathbf{I}$ is the deviatoric Cauchy stress resulting from the elastic distortions.

However, rather than use equation (3.8-101), we use

$$\dot{\gamma}_B = K_1 \left(\lambda_B^v - 1 + K_4 \right)^{K_2} \left(M_B \right)^{K_3}, \quad (3.8-102)$$

where $K_1 = \left(1 / \tau_{\text{base}} \right)^{K_3}$

$$M_B = \sqrt{\frac{1}{2} \mathbf{M}_{B_{\text{dev}}} : \mathbf{M}_{B_{\text{dev}}}}, \quad (3.8-103)$$

and $\mathbf{M}_{B_{\text{dev}}} = \mathbf{M}_B - \frac{1}{3} \text{tr}(\mathbf{M}_B) \mathbf{I}$ is the deviatoric Mandel stress.

This is consistent with the assumption of Mandel stress-driven inelastic behavior in ADINA's parallel-network framework, described in Section 3.17.2.

Further, unlike Bergström and Boyce (1998) who assume $\mathbf{l} = \mathbf{l}_B^e + \mathbf{D}_B^v$ where \mathbf{l} is the total spatial velocity gradient, and \mathbf{l}_B^e and \mathbf{D}_B^v are, respectively, the elastic velocity gradient and inelastic deformation rate for network B , we take the plastic spin to vanish $\mathbf{W}_B^v = \mathbf{0}$, and thus the relation (cf., equation 3.17-6)

$$\mathbf{l} = \mathbf{l}_B^e + \mathbf{X}_B^e \mathbf{D}_B^v \mathbf{X}_B^{e-1}, \quad (3.8-104)$$

is assumed. For more information about these kinematic assumptions, see Section 3.17.1.

Heat generation: A user-specified fraction of the energy dissipated by the viscous effects can be considered as heat generation. This heat generation can cause heating in a TMC (thermo-mechanical-coupling) analysis.

3.8.2.2.4 Constants and their interpretation

The Bergström-Boyce viscoelastic model is particularly powerful given that its flow rule in equation (3.8-102) requires only 3 physically interpretable material constants and one numerically required constant. Each parameter's corresponding designation in ADINA is given parenthetically in monospaced font and its dimensions in brackets, when applicable.

1. The constant $\tau_{base} > 0$ (TAUBASE) is the flow resistance of dimensions $\left[\frac{\text{M}}{\text{L T}^2} \right]$ (e.g., MPa). This parameter is not used when a value is entered for C1B.
2. The constant K_1 (C1B) is a positive viscous deformation rate constant of dimensions $\left[\frac{1}{\text{T}} \cdot \left(\frac{\text{M}}{\text{L T}^2} \right)^{-K_3} \right]$ (e.g., $\text{s}^{-1} \cdot (\text{MPa})^{-K_3}$). As $K_1 \rightarrow 0$, the viscous response vanishes, and the model recovers the material's equilibrium response (i.e., the special case of infinitely slow motions or load

application). Depending upon the consistent system of units in use, the numerical value of K_1 may be a very small number. For example, if the time unit is seconds, Pascals are used rather than MPa for stress, and K_3 is 5, then K_1 may be on the order of 1.0E-25 or smaller. This may result in numerical difficulties, in which case, a different consistent system of units should be used. For this reason, it is recommended to not enter a value for this parameter, but rather to specify the parameter τ_{base} (TAUBASE).

3. Constant $K_2 \in [-1, 0]$ (C2B) is the dimensionless chain stretch exponential. When $K_2 = 0$, the viscous response is insensitive to inelastic chain stretch.
 4. Constant K_3 (MMB) is a positive, dimensionless stress exponential which scales the effects of the deviatoric Mandel stress on inelastic flow. As $K_3 \rightarrow 0$, the viscous response becomes insensitive to the Mandel stress.
 5. Parameter K_4 (EOFFSET) is a small positive number (say, 1.0E-6) that prevents numerical difficulties as the inelastic chain stretch $\lambda_B^v \rightarrow 1$.
- In addition to those constants associated with equation (3.8-102), the Bergström-Boyce viscoelastic model requires the following parameters.
 1. RATIOBA: The ratio of hyperelastic contribution of network B to that of network A , as described in section 3.8.2.2.2.
 2. CALIBRATION: By default, ADINA assumes that the material parameters are calibrated in accordance to the Bergström-Boyce model defined in Bergström and Boyce (1998). When CALIBRATION=ADINA, no scaling of the viscous flow rate is performed. However, if the material parameters are calibrated using the MCalibration software developed by Veryst Engineering, then the user should specify CALIBRATION=MCAL. In this case, the viscous flow rate $\dot{\gamma}_B$ is automatically scaled by a factor of $\sqrt{2}^{K_3}$ thereby providing compatibility when using parameters generated by the

MCalibration software.

3. MAXITE: The Bergström-Boyce model requires an iterative material stress update to obtain the Mandel stress and inelastic distortion. The parameter MAXITE specifies the maximum number of iterations that ADINA should use for this stress update convergence.
4. RTOL: The tolerance required for convergence of the stress update. It is not recommended to loosen this tolerance from the default of 1.0E-12.

3.8.3 Mullins effect

When rubber is loaded to a given strain state, unloaded, then reloaded to the same strain state, the stress required for the reloading is less than the stress required for the initial loading. This phenomenon is referred to as the Mullins effect.

The Mullins effect can be included in the rubber-like materials. The material model used is the one described in the following reference:

- ref. R.W. Ogden and D. G. Roxburgh, "A pseudo-elastic model for the Mullins effect in filled rubber", *Proc. R. Soc. Lond. A* (1999) 455, 2861-2877.

We briefly summarize the main concepts below.

Fig 3.8-9 shows the Mullins effect in simple tension. On initial loading to force F_c , the specimen follows the force-deflection curve a-b-c. When the load is removed, the specimen follows the unloading curve c-d-a. On reloading to force F_c , the specimen follows the reloading curve a-d-c, and on further loading to force F_f , the specimen follows the loading curve c-e-f. When the load is removed, the specimen follows the unloading curve f-g-a, and, on reloading to force F_g , the specimen follows the reloading curve a-g-f.

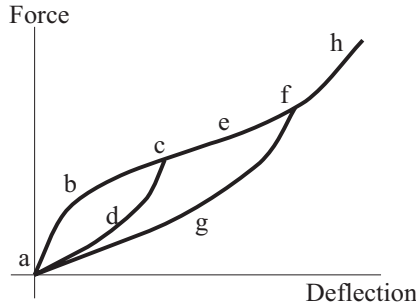


Figure 3.8-9: Mullins effect loading-unloading-reloading curves

Note that any permanent set associated with the Mullins effect is not included in the Ogden-Roxburgh model used here.

The Ogden-Roxburgh model, as implemented in ADINA, uses the following strain energy density expression:

$$\begin{aligned}\tilde{W} &= \eta W_D(\varepsilon_{ij}) + \phi(\eta), \text{ all except hyper-foam} \\ &= \eta W(\varepsilon_{ij}) + \phi(\eta), \text{ hyper-foam}\end{aligned}\quad (3.8-105a,b)$$

where $W(\varepsilon_{ij})$ is the total elastic strain energy density, $W_D(\varepsilon_{ij})$ is the deviatoric elastic strain energy density, η is an additional solution variable describing the amount of unloading and $\phi(\eta)$ is the damage function. \tilde{W} is referred to as the pseudo-energy function. In our implementation, the deviatoric strain energy density is used for the (almost) incompressible materials and the total strain energy density is used for compressible materials. For ease of writing, we discuss only the case of compressible materials; for incompressible materials, replace W by W_D in the equations below.

η is computed as

$$\eta = 1 - \frac{1}{r} \operatorname{erf} \left[\frac{1}{m} (W_m - W) \right] \quad (3.8-106)$$

where $\operatorname{erf}(x)$ is the error function

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp(-u^2) du \quad (3.8-107)$$

W_m is the maximum value of W encountered during the deformation history and m and r are material constants.

$\phi(\eta)$ is defined by

$$\frac{d\phi(\eta)}{d\eta} = -W \quad (3.8-108)$$

and is computed by numerical integration of $\dot{\phi}(\eta) = -W\dot{\eta}$. For a given value of W_m , there is a minimum value of η computed as

$$\eta_m = 1 - \frac{1}{r} \operatorname{erf} \left[\frac{W_m}{m} \right] \quad (3.8-109)$$

The value of $\phi(\eta)$ at $\eta = \eta_m$ is written $\phi(\eta_m)$. (Note: the subscript m in the term W_m means “maximum”, but the subscript m in the term η_m means “minimum”.) The time rate of change of $\phi(\eta_m)$ can be written as

$$\dot{\phi}(\eta_m) = (1 - \eta_m) \dot{W}_m = \frac{1}{r} \operatorname{erf} \left[\frac{W_m}{m} \right] \dot{W}_m \quad (3.8-110)$$

Physically, $\phi(\eta_m)$ is interpreted as the dissipation.

During loading, $W_m = W$, $\dot{W}_m > 0$ and $\eta = 1$. Therefore $\dot{\phi}(\eta) = 0$ and $\dot{\phi}(\eta_m) > 0$ during loading.

During unloading or reloading, $W_m > W$, $\dot{W}_m = 0$ and $\eta_m \leq \eta < 1$. Therefore $\dot{\phi}(\eta) \neq 0$ and $\dot{\phi}(\eta_m) = 0$ during unloading or reloading.

Material constants m and r do not have any direct physical significance. However Fig 3.8-10 shows the dependence of an unloading-reloading curve in simple tension on these parameters.

It is seen that, for an unloading-reloading loop in which $W_m \gg m$, the initial slope of the reloading curve is reduced by the factor $1 - \frac{1}{r}$. r must therefore be greater than 1.

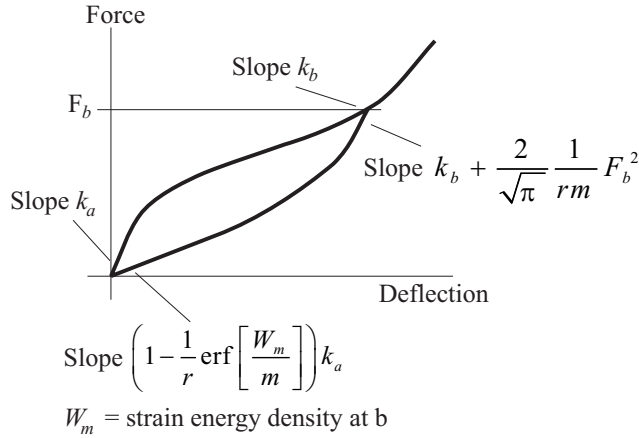


Figure 3.8-10: Dependence of reloading curve on Mullins effect material constants

It can also be shown that the dissipation of a loading-unloading cycle, as shown in Fig 3.8-11, can be written as

$$\phi = \int_A^E \sigma de = \frac{m}{r} \left[\frac{W_m}{m} \operatorname{erf} \left(\frac{W_m}{m} \right) - \frac{1}{\sqrt{\pi}} \left(1 - \exp \left(- \left(\frac{W_m}{m} \right)^2 \right) \right) \right] \quad (3.8-111)$$

where

$$W_m = \int_A^C \sigma de \quad (3.8-112)$$

${}_0\sigma$ is the engineering stress, e is the engineering strain. Therefore, given ϕ and W_m from two loading-unloading cycles of different

amplitude, m and r can be computed.

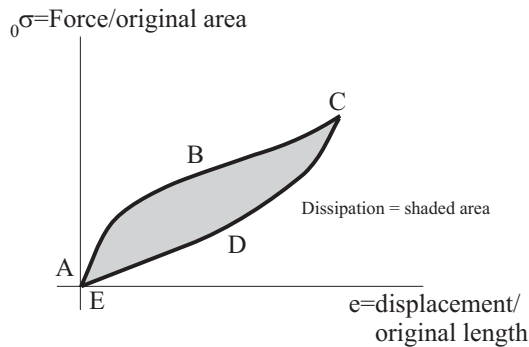


Figure 3.8-11: Dissipation in Mullins effect in a loading-reloading cycle

Heat generation: A user-specified fraction of the energy dissipated by the Mullins effect model can be considered as heat generation. This heat generation can cause heating in a TMC (thermo-mechanical-coupling) analysis.

Specification of input: To add Mullins effects to the rubber-like material model, you need to define the Mullins data using a data set of type rubber-Mullins, then specify the number of the rubber-Mullins data set in the rubber-like material model.

The rubber-Mullins data set includes:

The material constants r and m .

The heat generation factor (fraction of dissipation considered as heat generation). The default value is 0.

3.8.4 Orthotropic effect

A 2D network of orthotropic fibers can be included in the rubber-like material descriptions, see Figure 3.8-12. The directions of the fibers are described by the normal vectors \mathbf{n}_a , \mathbf{n}_b , with components $(n_a)_i$ and $(n_b)_i$ respectively. These normal vectors are defined in the undeformed configuration, and can change directions due to material deformations.

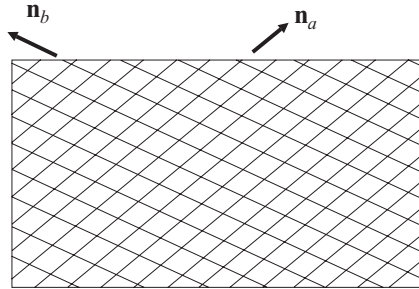


Figure 3.8-12: Network of orthotropic fibers

The orthotropic fibers are included by appending to the strain energy density the following additional term

$$W_O = \frac{k_1}{2k_2} \left\{ \exp \left[k_2 (J_4 - 1)^2 \right] + \exp \left[k_2 (J_6 - 1)^2 \right] - 2 \right\} \quad (3.8-113)$$

where $J_4 = I_4 I_3^{-1/3}$, $J_6 = I_6 I_3^{-1/3}$, $I_4 = C_{ij} (n_a)_i (n_a)_j$,

$I_6 = C_{ij} (n_b)_i (n_b)_j$, C_{ij} is the Cauchy-Green deformation tensor,

and k_1, k_2 are material constants. k_1 has the dimension of stress and k_2 is dimensionless. In the expression in brackets in (3.8-113), the first term represents the response of the \mathbf{n}_a fibers and the second term represents the response of the \mathbf{n}_b fibers.

This material description is described in the following reference:

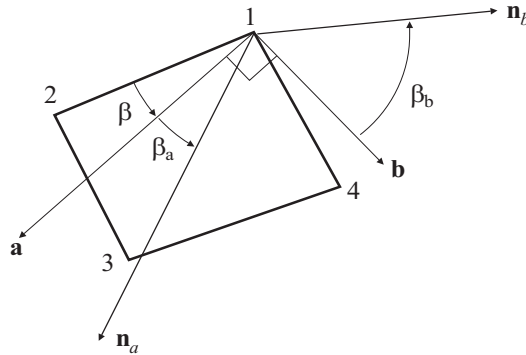
G. A. Holzapfel, T. C. Gasser, R. W. Ogden, "A New Constitutive Framework for Arterial Wall Mechanics and a Comparative Study of Material Models", *Journal of Elasticity*, 61: 1-48, 2000.

We note:

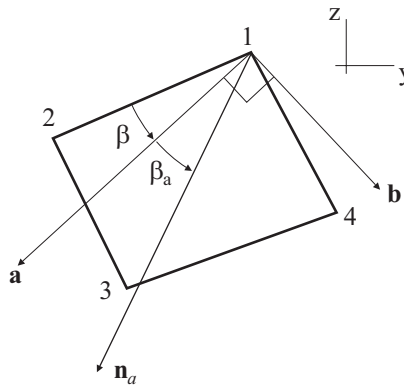
The normal directions \mathbf{n}_a and \mathbf{n}_b are defined as shown in Figure 3.8-13. In 2D analysis, the figures show how \mathbf{n}_a and \mathbf{n}_b are defined in terms of the input material axis angle β and the

offset angles β_a , β_b . \mathbf{n}_a and \mathbf{n}_b lie in the plane of the element. In axisymmetric analysis, it is also allowed to set \mathbf{n}_b to the hoop (x) direction. In 3D analysis, the figure shows how \mathbf{n}_a and \mathbf{n}_b are defined in terms of the material directions \mathbf{a} and \mathbf{b} . \mathbf{n}_a and \mathbf{n}_b lie in the plane defined by \mathbf{a} and \mathbf{b} .

\mathbf{n}_a and \mathbf{n}_b need not be perpendicular to each other.

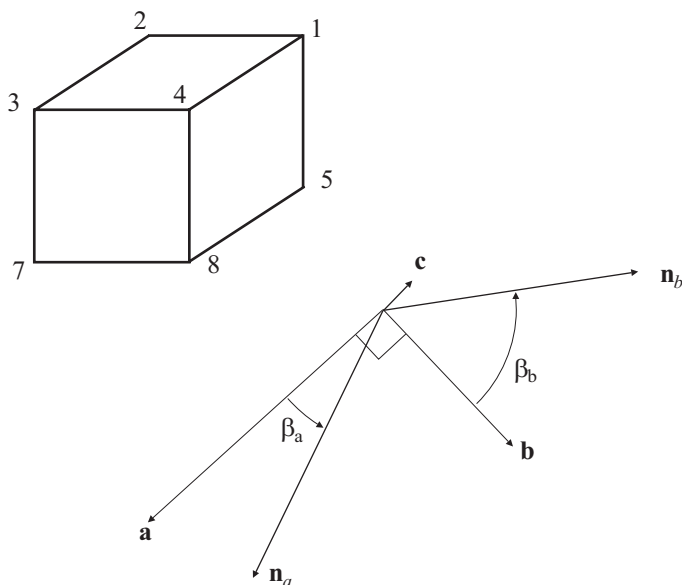


a) 2D element, hoop option not used



b) 2D element, hoop option used, $\mathbf{n}_b = x$

Figure 3.8-13: Orientation of orthotropic axes



c) 3D element, \mathbf{n}_a , \mathbf{n}_b lie in the \mathbf{a} - \mathbf{b} plane

Figure 3.8-13: (continued)

I_4 is interpreted as the square of the stretch of the fiber initially orientated in direction \mathbf{n}_a ; I_6 is interpreted as the square of the stretch of the fiber initially orientated in direction \mathbf{n}_b ; see Ref KJB, Example 6.7.

J_4 is interpreted as the deviatoric part of I_4 ; J_6 is interpreted as the deviatoric part of I_6 . It is assumed that the material is nearly incompressible.

When $J_4 < 1$, the material is in compression in direction \mathbf{n}_a ; when $J_6 < 1$, the material is in compression in direction \mathbf{n}_b . It is possible to remove the orthotropic effect when the material is in compression in either of the fiber directions.

Specification of input: To add orthotropic effects to the rubber-like material model, you need to define the orthotropic data using a data set of type rubber-orthotropic, then include the number of the rubber-orthotropic data set in the rubber-like material model description.

The rubber-orthotropic data set includes:

- ▶ The material offset angles β_a, β_b (default values are 0).
- ▶ The material constants k_1, k_2 .
- ▶ A flag indicating whether the orthotropic effect is removed when the material is in compression in either of the fiber directions (the default is to remove the orthotropic effect when the material is in compression).
- ▶ A flag indicating whether direction \mathbf{n}_b is the x (hoop) direction (the default is that \mathbf{n}_b is not the x (hoop) direction).

3.8.5 Thermal strain effect

When the material is temperature-dependent, you can include a coefficient of thermal expansion. The coefficient of thermal expansion is $\alpha(\theta)$, where θ is the temperature. Notice that the coefficient of thermal expansion is temperature dependent.

The thermal strain is calculated as

$$e_{th} = \alpha(\theta)(\theta - \theta_{REF}) - \alpha(^0\theta)(^0\theta - \theta_{REF}) \quad (3.8-114)$$

where $^0\theta$ is the initial temperature and θ_{REF} is the material reference temperature. This is the same formula as is used for the other thermo-elastic materials, see Section 3.5.

The thermal strain is assumed to be isotropic.

When the thermal strain is non-zero, the deformation gradient \mathbf{X} is assumed to be decomposed into a thermal deformation gradient \mathbf{X}_{th} and a mechanical deformation gradient \mathbf{X}_m , using

$$\mathbf{X} = \mathbf{X}_m \mathbf{X}_{th} \quad (3.8-115)$$

The thermal deformation gradient is

$$\mathbf{X}_{th} = (1 + e_{th})\mathbf{I} \quad (3.8-116)$$

therefore the mechanical deformation gradient is

$$\mathbf{X}_m = (1 + e_{th})^{-1} \mathbf{X} \quad (3.8-117)$$

the mechanical Cauchy-Green deformation tensor is

$$\mathbf{C}_m = (1 + e_{th})^{-2} \mathbf{C} \quad (3.8-118)$$

and the mechanical Green-Lagrange strain tensor is

$$\boldsymbol{\varepsilon}_m = (1 + e_{th})^{-2} \boldsymbol{\varepsilon} - \frac{1}{2} (1 - (1 + e_{th})^{-2}) \mathbf{I} \quad (3.8-119)$$

For small thermal strains, (3.8-119) reduces to $\boldsymbol{\varepsilon}_m \approx \boldsymbol{\varepsilon} - e_{th} \mathbf{I}$, so that the strains are nearly the sum of the mechanical and thermal strains, as in small strain analysis. However, we do not assume that the thermal strains are small.

The strain energy densities are computed using the mechanical deformations. This is done by computing all invariants and stretches using the mechanical deformations, e.g. the mechanical Cauchy-Green deformation tensor.

The 2nd Piola-Kirchhoff stresses are obtained by differentiating the strain energy density with respect to the total strains. Since the strain energy density is a function of the mechanical strains, we obtain

$$\begin{aligned} S_{ij} &= \frac{1}{2} \left(\frac{\partial W}{\partial (\boldsymbol{\varepsilon})_{ij}} + \frac{\partial W}{\partial (\boldsymbol{\varepsilon})_{ji}} \right) \\ &= \frac{1}{2} \left(\frac{\partial W}{\partial (\boldsymbol{\varepsilon}_m)_{ab}} \frac{\partial (\boldsymbol{\varepsilon}_m)_{ab}}{\partial (\boldsymbol{\varepsilon})_{ij}} + \frac{\partial W}{\partial (\boldsymbol{\varepsilon}_m)_{ba}} \frac{\partial (\boldsymbol{\varepsilon}_m)_{ba}}{\partial (\boldsymbol{\varepsilon})_{ji}} \right) \\ &= \left((1 + e_{th})^{-2} \right) \frac{1}{2} \left(\frac{\partial W}{\partial (\boldsymbol{\varepsilon}_m)_{ij}} + \frac{\partial W}{\partial (\boldsymbol{\varepsilon}_m)_{ji}} \right) \end{aligned} \quad (3.8-120)$$

With this definition, the 2nd Piola-Kirchhoff stresses are conjugate to the Green-Lagrange strains.

3.8.6 TRS material

The rubber-like materials can include the TRS (thermorheologically simple) material assumption. In this assumption, the material properties are independent of temperature, but the viscoelastic time constants of the Holzapfel viscoelastic model are functions of temperature as described in the viscoelastic material effect (time-temperature superposition).

A TRS material requires a rubber-table of type TRS. The number of the rubber-table data set is specified as part of the rubber-like material description.

The rubber-table of type TRS data set is a table with a different temperature for each row of the table. Each row of the table contains the coefficient of thermal expansion (enter 0 if there are no thermal effects).

We do not include the absolute temperature in the strain energy density function, so we are not capturing the Gough-Joule thermoelastic effects, described in

ref. Gurtin, Fried, Anand, *The Mechanics and Thermodynamics of Continua*, Cambridge University Press, 2010, Section 58.

3.8.7 Temperature-dependent material

The rubber-like materials can be fully temperature-dependent.

Specification of input: The input of the material properties for a fully temperature-dependent material is done using a rubber-table. The rubber-table has different forms depending upon whether the Mooney-Rivlin, Ogden, Arruda-Boyce, hyper-foam, Sussman-Bathe or eight-chain relationships are used, hence there is a rubber-table of type Mooney-Rivlin, a rubber-table of type Ogden, etc. For brevity, we discuss only the case of a Mooney-Rivlin material.

A Mooney-Rivlin material that is fully temperature-dependent requires a rubber-table of type Mooney-Rivlin. The number of the rubber-table data set is specified as part of the Mooney-Rivlin material description.

The rubber-table of type Mooney-Rivlin data set is a table with a different temperature for each row of the table. Each row of the table contains

- ▶ The coefficient of thermal expansion.
- ▶ The material properties (C_1 to C_9 , constants D_1 to D_2 and the bulk modulus κ).
- ▶ The curve-fitting number (0 if curve-fitting is not used).
- ▶ The rubber-viscoelastic data set number, if the model includes viscoelastic effects.
- ▶ The rubber-Mullins data set number, if the model includes Mullins effects.
- ▶ The rubber-orthotropic data set number, if the model includes orthotropic effects.

You can specify a different curve-fitting number for each temperature.

You can specify a different rubber-viscoelastic data set number, rubber-Mullins data set number or rubber-orthotropic data set number for each temperature. In this way, the material constants for the viscoelastic, Mullins or orthotropic effects can be temperature-dependent. However, if one row of the rubber-table has viscoelastic data, all rows must have viscoelastic data. Similarly, if one row of the rubber-table has Mullins or orthotropic data, all rows must have this data.

Solution process: The strain energy density is computed by interpolation from the strain energy densities of the bounding temperatures as follows: Let θ be the current temperature, θ_1 be the temperature in the rubber-table just below the current temperature and θ_2 be the temperature in the rubber-table just above the current temperature. Then the program computes the strain energy densities W_1 and W_2 , and the derivatives of the strain energy densities, using the material properties for θ_1 and θ_2 respectively. Finally the program computes the strain energy density and its derivatives using linear interpolation, for example

$$W = \frac{\theta_2 - \theta}{\theta_2 - \theta_1} W_1 + \frac{\theta - \theta_1}{\theta_2 - \theta_1} W_2 \quad (3.8-121)$$

3.8.8 Rubber stability indicators

In this section we give the theory underlying the stability indicators for the rubber-like material models. The stability indicators can be plotted in the AUI, using the command

```
MATERIALSHOW STRAIN CURVETYPE=STABILITY
```

3.8.8.1 Introduction

We consider the stability of a unit cube of incompressible hyperelastic material. In this investigation, we only consider homogeneous deformations, and only stability considering perturbations that are also homogeneous.

The cube is subjected to applied loads R_1 , R_2 , R_3 (Fig 3.8-14). These loads are deformation-independent. Since the cube is a unit cube, these loads can also be interpreted as engineering stresses.

The corresponding displacements of the cube are u_1 , u_2 , u_3 . The stretches and true strains corresponding to these displacements are $\lambda_i = 1 + u_i$ and $e_i = \ln \lambda_i$, again using that the cube is a unit cube.

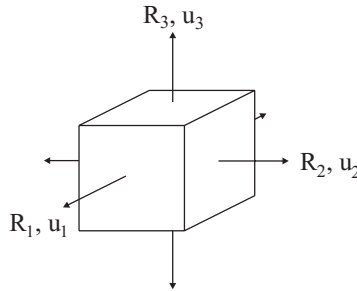


Figure 3.8-14 Unit cube under applied loads

The strain energy of the cube is $W = W(e_1, e_2, e_3)$, with different strain energy functions for different material models (e.g. Mooney-Rivlin, Ogden, Arruda-Boyce, Sussman-Bathe, eight-chain).

Since the material is incompressible, we must enforce the condition $e_1 + e_2 + e_3 = 0$.

Also note, in the derivation below, the summation convention is not used.

Equilibrium

Equilibrium, along with incompressibility, is determined using a Lagrange multiplier approach:

$$\hat{W} = W + m(e_1 + e_2 + e_3) \quad (3.8-122)$$

where m is the Lagrange multiplier. Taking variations of (3.8-122) gives

$$\delta \hat{W} = \sum_i \left(\frac{\partial W}{\partial e_i} + m \right) \delta e_i + (e_1 + e_2 + e_3) \delta m \quad (3.8-123)$$

The virtual work done by the loads is

$$\delta R = \sum_i R_i \delta u_i = \sum_i R_i \lambda_i \delta e_i \quad (3.8-124)$$

and equating the external and internal virtual work gives

$$\frac{\partial W}{\partial e_i} + m = R_i \lambda_i, \text{ no sum on } i, \quad (3.8-125a)$$

$$e_1 + e_2 + e_3 = 0 \quad (3.8-125b)$$

3.8.8.2 Stability

The stability criterion is obtained taking increments of the loads and determining the corresponding increments of the displacements. The resulting matrix equation is

$$\begin{bmatrix} \frac{\partial^2 W}{\partial e_1^2} - R_1 \lambda_1 & \frac{\partial^2 W}{\partial e_1 \partial e_2} & \frac{\partial^2 W}{\partial e_1 \partial e_3} & 1 \\ & \frac{\partial^2 W}{\partial e_2^2} - R_2 \lambda_2 & \frac{\partial^2 W}{\partial e_2 \partial e_3} & 1 \\ & & \frac{\partial^2 W}{\partial e_3^2} - R_3 \lambda_3 & 1 \\ \text{symm} & & & 0 \end{bmatrix} \begin{bmatrix} \Delta e_1 \\ \Delta e_2 \\ \Delta e_3 \\ \Delta m \end{bmatrix} = \begin{bmatrix} \lambda_1 \Delta R_1 \\ \lambda_2 \Delta R_2 \\ \lambda_3 \Delta R_3 \\ 0 \end{bmatrix} \quad (3.8-126)$$

Equation (3.8-126) can of course be solved for Δe_i and Δm given ΔR_i , but this does not give insight into the stability. The eigenvalues and eigenvectors of the coefficient matrix could be solved for, but the eigenvectors need not satisfy incompressibility.

To continue, we note that the stability matrix equation (3.8-126) has the form

$$\begin{bmatrix} & & & 1 \\ & \mathbf{KEE} & & 1 \\ & & & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} \Delta e_1 \\ \Delta e_2 \\ \Delta e_3 \\ \Delta m \end{bmatrix} = \begin{bmatrix} \Delta r_1 \\ \Delta r_2 \\ \Delta r_3 \\ 0 \end{bmatrix} \quad (3.8-127)$$

where \mathbf{KEE} is a symmetric 3x3 matrix. We now choose a new basis for the strain increments:

$$\begin{bmatrix} \Delta e_1 \\ \Delta e_2 \\ \Delta e_3 \end{bmatrix} = \mathbf{A} \begin{bmatrix} \Delta y_1 \\ \Delta y_2 \\ \Delta y_3 \end{bmatrix} \quad (3.8-128)$$

$$\text{where } \mathbf{A} = \begin{bmatrix} \sqrt{\frac{2}{3}} & 0 & \frac{1}{\sqrt{3}} \\ -\frac{1}{2}\sqrt{\frac{2}{3}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} \\ -\frac{1}{2}\sqrt{\frac{2}{3}} & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} \end{bmatrix}. \quad \mathbf{A} \text{ is orthogonal with}$$

determinant=1. We see that

$$\begin{bmatrix} \Delta y_1 \\ \Delta y_2 \\ \Delta y_3 \end{bmatrix} = \mathbf{A}^T \begin{bmatrix} \Delta e_1 \\ \Delta e_2 \\ \Delta e_3 \end{bmatrix} = \begin{bmatrix} \sqrt{\frac{2}{3}} & -\frac{1}{2}\sqrt{\frac{2}{3}} & -\frac{1}{2}\sqrt{\frac{2}{3}} \\ 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \end{bmatrix} \begin{bmatrix} \Delta e_1 \\ \Delta e_2 \\ \Delta e_3 \end{bmatrix} \quad (3.8-129)$$

Note that Δy_1 and Δy_2 are insensitive to the volume change and Δy_3 is proportional to the volume change $\Delta e_1 + \Delta e_2 + \Delta e_3$.

In the new basis, the stability matrix equation has the form

$$\begin{bmatrix} & 0 \\ \mathbf{KYY} & 0 \\ & \sqrt{3} \\ 0 & 0 & \sqrt{3} & 0 \end{bmatrix} \begin{bmatrix} \Delta y_1 \\ \Delta y_2 \\ \Delta y_3 \\ \Delta m \end{bmatrix} = \begin{bmatrix} \Delta s_1 \\ \Delta s_2 \\ \Delta s_3 \\ 0 \end{bmatrix} \quad (3.8-130)$$

where $\mathbf{KYY} = \mathbf{A}^T \mathbf{KEE} \mathbf{A}$ and $\begin{bmatrix} \Delta s_1 \\ \Delta s_2 \\ \Delta s_3 \end{bmatrix} = \mathbf{A}^T \begin{bmatrix} \Delta r_1 \\ \Delta r_2 \\ \Delta r_3 \end{bmatrix}$.

Equation (3.8-130) is solved noting that the 4th row equation is satisfied by the choice $\Delta y_3 = 0$; thus the matrix stability equation reduces to

$$\begin{bmatrix} KYY_{11} & KYY_{12} & 0 \\ KYY_{12} & KYY_{22} & 0 \\ KYY_{13} & KYY_{23} & \sqrt{3} \end{bmatrix} \begin{bmatrix} \Delta y_1 \\ \Delta y_2 \\ \Delta m \end{bmatrix} = \begin{bmatrix} \Delta s_1 \\ \Delta s_2 \\ \Delta s_3 \end{bmatrix} \quad (3.8-131)$$

and (3.8-131) is solved by

$$\begin{bmatrix} KYY_{11} & KYY_{12} \\ KYY_{12} & KYY_{22} \end{bmatrix} \begin{bmatrix} \Delta y_1 \\ \Delta y_2 \end{bmatrix} = \begin{bmatrix} \Delta s_1 \\ \Delta s_2 \end{bmatrix} \quad (3.8-132a)$$

$$\Delta m = \frac{1}{\sqrt{3}} (\Delta s_3 - KYY_{13} \Delta y_1 - KYY_{23} \Delta y_2) \quad (3.8-132b)$$

Thus we conclude that stability is governed by the eigenvalues of $\begin{bmatrix} KYY_{11} & KYY_{12} \\ KYY_{12} & KYY_{22} \end{bmatrix}$, and the stability criterion is the smallest eigenvalue of this matrix.

3.8.8.3 Stability in uniaxial tension

In uniaxial tension, $R_2 = R_3 = 0$, so, from equilibrium, $m = -\frac{\partial W}{\partial e_2}$

and $R_1 \lambda_1 = \frac{\partial W}{\partial e_1} - \frac{\partial W}{\partial e_2}$. Also, at equilibrium, $e_2 = e_3 = -\frac{1}{2} e_1$.

However, there are still three admissible strain increments Δe_1 , Δe_2 , Δe_3 , so the 4x4 matrix stability equations given above ((3.8-126) to (3.8-132)) are applicable.

3.8.8.4 Stability in pure shear

In pure shear, $e_2 = 0$, hence $\Delta e_2 = 0$ and there are only two admissible strain increments. Here $R_3 = 0$, so, from equilibrium,

$m = -\frac{\partial W}{\partial e_3}$ and $R_1 \lambda_1 = \frac{\partial W}{\partial e_1} - \frac{\partial W}{\partial e_3}$. Also, at equilibrium, $e_3 = -e_1$.

The matrix stability equation becomes

$$\begin{bmatrix} \frac{\partial^2 W}{\partial e_1^2} - R_1 \lambda_1 & \frac{\partial^2 W}{\partial e_1 \partial e_3} & 1 \\ & \frac{\partial^2 W}{\partial e_3^2} & 1 \\ \text{symm} & & 0 \end{bmatrix} \begin{bmatrix} \Delta e_1 \\ \Delta e_3 \\ \Delta m \end{bmatrix} = \begin{bmatrix} \Delta r_1 \\ \Delta r_3 \\ 0 \end{bmatrix} \quad (3.8-133)$$

This is solved using the change of basis

$$\begin{bmatrix} \Delta e_1 \\ \Delta e_3 \end{bmatrix} = \mathbf{A} \begin{bmatrix} \Delta y_1 \\ \Delta y_2 \end{bmatrix} \quad (3.8-134)$$

where now $\mathbf{A} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$, so that the matrix stability equation (3.8-133) becomes

$$\begin{bmatrix} \mathbf{KYY} & 0 \\ & \sqrt{2} \\ 0 & \sqrt{2} & 0 \end{bmatrix} \begin{bmatrix} \Delta y_1 \\ \Delta y_2 \\ \Delta m \end{bmatrix} = \begin{bmatrix} \Delta s_1 \\ \Delta s_2 \\ 0 \end{bmatrix} \quad (3.8-135)$$

and (3.8-135) is solved using

$$KYY_{11}\Delta y_1 = \Delta s_1, \Delta y_2 = 0, \Delta m = \frac{1}{\sqrt{2}}(\Delta s_2 - KYY_{12}\Delta y_1) \quad (3.8-136a,b,c)$$

Stability is governed by coefficient KYY_{11} . (Note, this is a different KYY_{11} than is used for the cases of 3 admissible strain increments.)

3.8.8.5 Stability in equibiaxial tension

In equibiaxial tension, $R_1 = R_2$ and $R_3 = 0$, so, from equilibrium,

$$m = -\frac{\partial W}{\partial e_3} \text{ and } R_1\lambda_1 = R_2\lambda_2 = \frac{\partial W}{\partial e_1} - \frac{\partial W}{\partial e_3}. \text{ Also, at equilibrium}$$

$e_2 = e_1$ and $e_3 = -2e_1$. Since there are still three admissible strain increments, $\Delta e_1, \Delta e_2, \Delta e_3$, the 4x4 matrix stability equations given above ((3.8-126) to (3.8-132)) are applicable.

3.8.8.6 Comments

The results from the stability indicators can be checked by running

a corresponding ADINA finite element model, using dead loads. When the stability indicator is positive, the stiffness matrix has all positive pivots; when the stability indicator is negative, the stiffness matrix has at least one negative pivot.

3.9 Geotechnical material models

3.9.1 Curve description material model

- The curve description model can be employed with the **2-D solid** (plane strain and axisymmetric) and **3-D solid** elements.
- The curve description model can be used with the **small displacement** and **large displacement** formulations. In all cases, small strains are assumed.

When used with the small displacement formulation, a materially-nonlinear-only formulation is employed, and when used with the large displacement formulation, a TL formulation is employed.

- The curve description model is a simple incremental stress-strain law used to represent the response of geological materials. The model describes the instantaneous bulk and shear moduli as piecewise linear functions of the current volumetric strain, as shown in Fig. 3.9-1. An explicit yield condition is not used and whether the material is loading or unloading is determined by the history of the volumetric strain only.

- To present the governing constitutive relations, let

${}^t e_{ij}$ = total strains (the left superscript "t" always refers to time t)

e_{ij} = incremental strains

${}^t e_m = \sum_i \frac{{}^t e_{ii}}{3}$ = mean strain (negative in compression)

$e_m = \sum_i \frac{e_{ii}}{3}$ = incremental mean strain

${}^t g_{ij} = {}^t e_{ij} - {}^t e_m \delta_{ij}$ = deviatoric strains

g_{ij} = incremental deviatoric strains

${}^t\sigma_{ij}$ = total stresses (negative in compression)

σ_{ij} = incremental stresses

${}^t\sigma_m = \sum_i \frac{{}^t\sigma_{ii}}{3}$ = mean stress

$\sigma_m = \sum_i \frac{\sigma_{ii}}{3}$ = incremental mean stress

p_{min} = minimum mean stress ever reached

${}^t s_{ij} = {}^t\sigma_{ij} - {}^t\sigma_m \delta_{ij}$ = deviatoric stresses

s_{ij} = incremental deviatoric stresses

${}^tG, {}^tK$ = shear and bulk moduli

The incremental stress-strain relations using the curve description model are then

$$s_{ij} = 2 {}^tG g_{ij}$$

$$\sigma_m = 3 {}^tK e_m$$

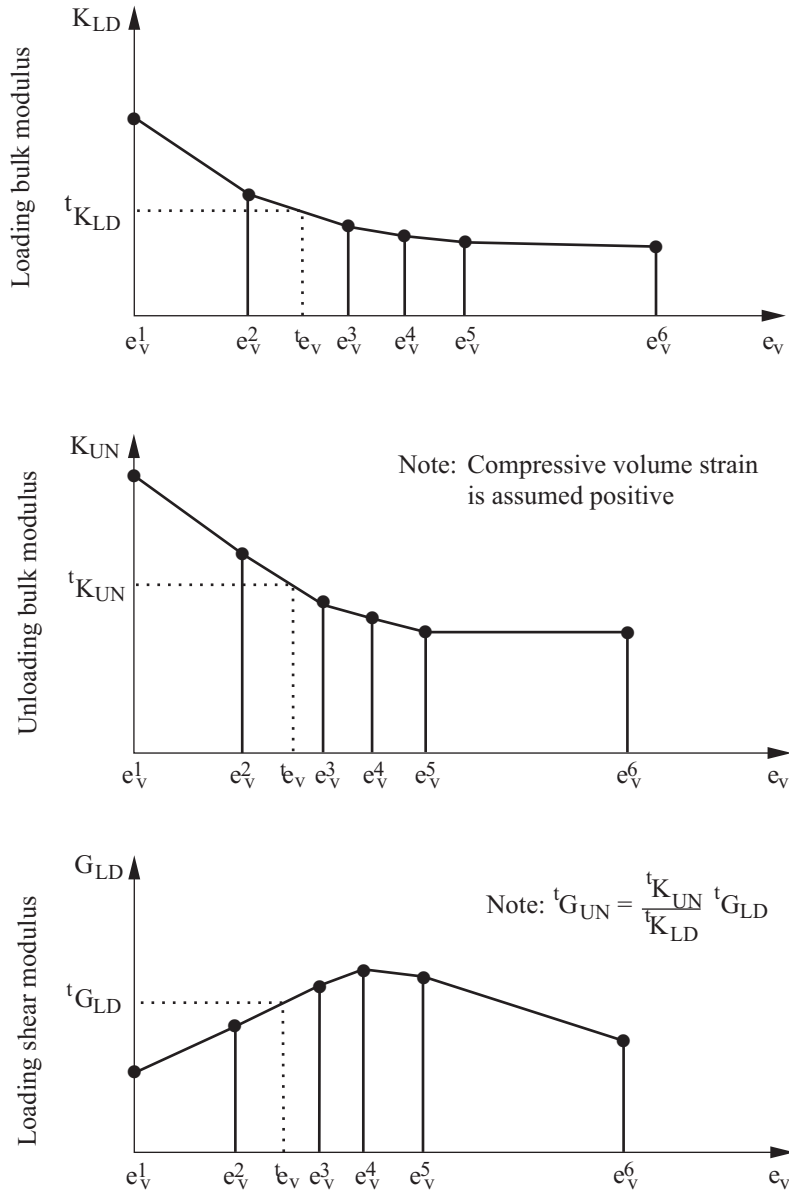


Figure 3.9-1: Input of the curve description model

- The instantaneous bulk and shear moduli, tK and tG , are functions of the loading condition, and the volumetric strain te_v is defined as

$${}^te_v = e_{grav} + (-3{}^te_m)$$

where e_{grav} is the volumetric strain (three times the mean strain and taken positive in compression) due to the gravity pressure and te_m is the mean strain at time t . Defining e_{min} as the minimum mean strain ever reached during the solution, we have that the material is loading if ${}^te_m \leq e_{min}$ and the material is unloading if ${}^te_m > e_{min}$ i.e.,

$${}^tK = \begin{cases} {}^tK_{LD} & \text{when } {}^te_m \leq e_{min} \\ {}^tK_{UN} & \text{when } {}^te_m > e_{min} \end{cases}$$

and

$${}^tG = \begin{cases} {}^tG_{LD} & \text{when } {}^te_m \leq e_{min} \\ {}^tG_{UN} & \text{when } {}^te_m > e_{min} \end{cases}$$

- Note that the loading conditions for both the bulk and the shear moduli are determined by the history of te_m only. The values of ${}^tK_{LD}$, ${}^tK_{UN}$, and ${}^tG_{LD}$ are obtained using the curves in Fig. 3.9-1, and the modulus ${}^tG_{UN} = {}^tG_{LD} \left(\frac{{}^tK_{UN}}{{}^tK_{LD}} \right)$.

- The incremental solution at time t is obtained using the equations

$${}^{t+\Delta t}s_{ij} = 2 {}^tG \left({}^{t+\Delta t}g_{ij} - {}^tg_{ij} \right) + {}^ts_{ij}$$

and

$${}^{t+\Delta t}\sigma_{ij} = 3 {}^tK \left({}^{t+\Delta t}e_{ij} - {}^te_{ij} \right) + {}^t\sigma_{ij}$$

To obtain tG and tK , we check whether the loading or unloading conditions are active by comparing the current volumetric strains and previous volumetric strains. To start the procedure at time 0,

loading conditions are assumed. It should be noted that the stresses at time $t + \Delta t$ are calculated using the material moduli pertaining to time t .

- An important additional analysis option is that the material can weaken under loading conditions if tensile stresses exceed preassigned values. Since the curve description model has been developed primarily for the analysis of geological materials, the material weakening is assumed to occur once the principal tensile stresses due to the applied loading exceed the in-situ gravity pressure p (taken positive). The gravity pressure p is calculated as

$$p = \sum_{i=1}^N h_i p_i \quad \text{where } h_i \text{ are the element interpolation functions and}$$

p_i is the pressure at the element nodes. The nodal pressure p_i is calculated as $p_i = \gamma \cdot Z_i$ where γ is the material density and Z_i the nodal Z coordinate, which coincides with the vertical direction. The material weakening can be included using a *tension cut-off* model or a *cracking* model.

- In both modes of behavior (tension cut-off and cracking) the principal stresses due to the external loading are calculated and compared with the in-situ gravity pressure. Once the principal tensile stress is equal to the in-situ gravity pressure, the material is treated as being orthotropic, with the modulus corresponding to the direction of the principal tensile stress being multiplied by a stiffness reduction factor (an input parameter). Another factor, also an input parameter, is applied to reduce the shear stiffness.

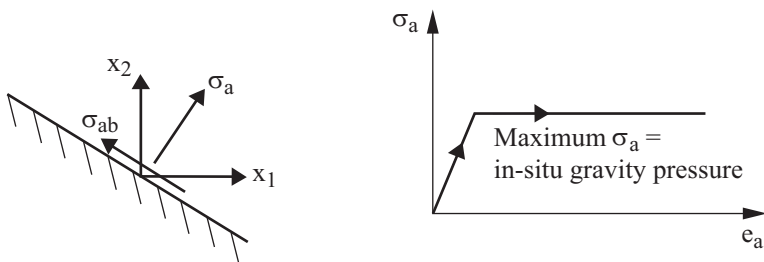
- In the tension cut-off mode, the normal and shear stiffnesses corresponding to the direction of the maximum principal tensile stress are reduced, but the stresses are fully retained. The model therefore simulates elastic-plastic flow of the material.

On the other hand, in the cracking mode the stiffnesses are reduced, and in addition the normal and shear stresses due to external loads and corresponding to the direction of stiffness reduction are released. The reduction of stiffness together with the stress release models a crack, i.e., a tensile failure plane, that forms at right angles to the direction of the maximum principal tensile stress.

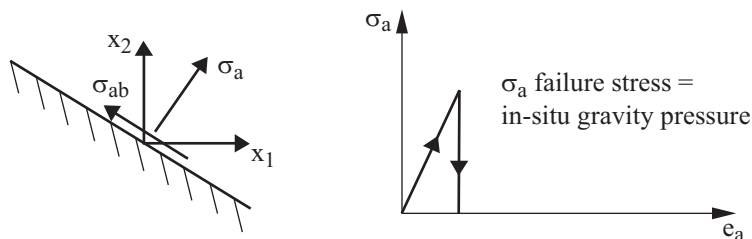
- In subsequent load steps, tension cut-off or cracking only in the direction already determined and/or other(s) perpendicular to it may be active, i.e., no change in the tension cut-off or cracking direction is considered. The tension cut-off or cracking mode is considered to be inactive provided the strain normal to the crack (in the direction of tension cut-off) becomes both negative and less than the strain at which the failure occurred initially; otherwise it remains active.

Fig. 3.9-2 illustrates the tension cut-off and cracking options.

- Note that the Z direction should be the vertical direction and that the ground level is assumed to be at $Z = 0$ in the calculation of the in-situ gravity pressure.



(a) Option of tension cut-off



(b) Option of cracking

ADINA checks for tension cut-off or cracking at each integration point.

Figure 3.9-2: Tension cut off and cracking options

- The in-situ gravity pressure is evaluated using the input for material density which corresponds to the weight per unit volume. This is different than the mass density which is used in the element mass matrix calculation.

3.9.2 Drucker-Prager material model

- The Drucker-Prager model is based on
 - ▶ The Drucker-Prager yield condition (see p. 604, ref. KJB)
 - ▶ A non-associative flow rule using the Drucker-Prager and cap yield functions
 - ▶ A perfectly-plastic Drucker-Prager yield behavior
 - ▶ Tension cut-off
 - ▶ Cap hardening
- The Drucker-Prager model can be used with the **2-D solid** and **3-D solid** elements.
- The Drucker-Prager model can be used with the **small displacement/small strain** and **large displacement/small strain** formulations.

When used with the small displacement/small strain formulation, a materially-nonlinear-only formulation is employed, and when used with the large displacement/small strain formulation, a TL formulation is employed.
- Fig. 3.9-3 summarizes some important features of the Drucker-Prager model (with tension cut-off and cap hardening).

The Drucker-Prager yield function is given by:

$${}^t f_{DP} = \alpha {}^t J_1 + \sqrt{{}^t J_{2D}} - k$$

where α and k are material parameters and are functions of the friction angle ϕ and cohesion coefficient c , and ${}^t J_1$ and ${}^t J_{2D}$ are the first stress invariant and the second deviatoric stress invariant,

respectively.

The corresponding potential function is given by

$${}^t g_{DP} = \beta {}^t J_1 + \sqrt{{}^t J_{2D}} - u$$

where β is a function of the dilatation angle ψ , u is not required as input parameter.

The parameters α , k and β must be properly defined. They are dependent on the type of application to be modeled. In particular, they can be determined by matching the Drucker-Prager criterion to the Mohr-Coulomb criterion (see Section 3.9.4 and Figs. 3.9-6 through 3.9-9 for details).

A compressive meridian match gives:

$$\alpha = \frac{2 \sin \phi}{\sqrt{3} (3 - \sin \phi)} \quad k = \frac{6c \cos \phi}{\sqrt{3} (3 - \sin \phi)} \quad \beta = \frac{2 \sin \psi}{\sqrt{3} (3 - \sin \psi)}$$

A tensile meridian match gives:

$$\alpha = \frac{2 \sin \phi}{\sqrt{3} (3 + \sin \phi)} \quad k = \frac{6c \cos \phi}{\sqrt{3} (3 + \sin \phi)} \quad \beta = \frac{2 \sin \psi}{\sqrt{3} (3 + \sin \psi)}$$

A plane strain match gives:

$$\alpha = \frac{\tan \phi}{\sqrt{9 + 12 \tan^2 \phi}} \quad k = \frac{3c}{\sqrt{9 + 12 \tan^2 \phi}}$$

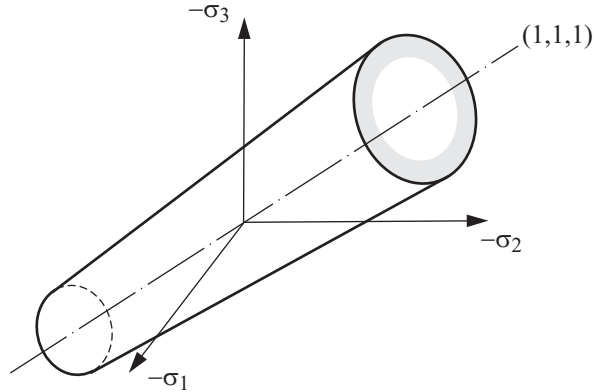
The cap yield function depends on the shape of the cap. For a plane cap:

$${}^t f_C = -{}^t J_1 + {}^t J_1^a$$

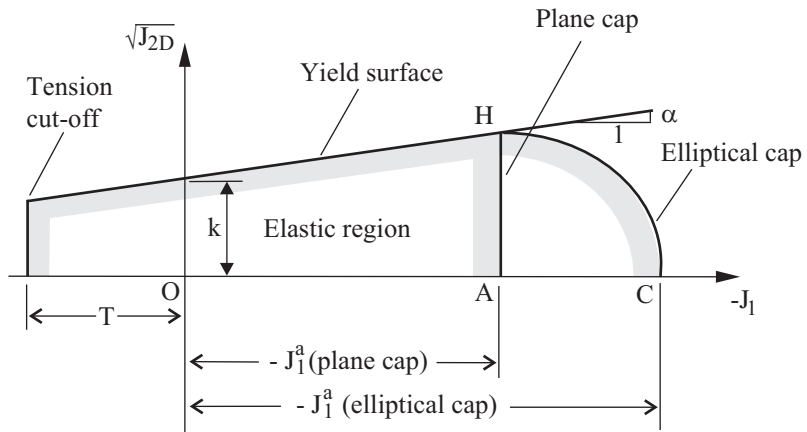
where ${}^t J_1^a$ is a function of the volumetric plastic strain ${}^t e_V^P$:

$${}^t J_1^a = -\frac{1}{D} \ln \left(1 - \frac{{}^t e_V^P}{W} \right) + {}^0 J_1^a$$

where ${}^0 J_1^a$ is the cap initial position and W and D are material constants.



a) Drucker-Prager yield surface in principal stress space



b) Drucker-Prager model in J_1 versus $\sqrt{J_2}D$ space

Figure 3.9-3: Drucker-Prager model

For an elliptical cap:

$${}^t f_C = \left({}^t J_1 + {}^t L \right)^2 + R^2 \left({}^t J_{2D} - {}^t B^2 \right)$$

where ${}^t B$ is the vertical semi-axis of the ellipse (AH), R is the cap ratio (AC/AH) and ${}^t L$ is equal to OA. The dependence of ${}^0 J_1^a$ on the volumetric plastic strain is the same as for the plane cap. In the case of tension cut-off, T is the maximum value that ${}^t J_1^a$ can take.

- We note that:
 - ▶ Unlike for the von Mises model, the Drucker-Prager yield function cannot be used with hardening (the material is always assumed to be elastic-perfectly-plastic, except for cap hardening).
 - ▶ If $\alpha(\phi)$ approaches zero (minimum value of α is taken to be 10^{-5}), the initial position of the cap ${}^0 J_1^a$ is moved far to the right in Figure 3.9-3 and is not reached in the analysis, and the position of the tension cut-off T is moved far to the left and is also not reached in the analysis, then the Drucker-Prager yield condition approaches the von Mises elastic-perfectly-plastic yield condition.
 - ▶ In the case of Drucker-Prager yielding, the dilatency (volume expansion of the material in shear) is governed by the magnitude of $\beta(\psi)$ (for the von Mises yield condition $\beta=0$ and there is no dilatency).
 - ▶ Cap yielding leads to an increase of compressive plastic volumetric strain. If the plane cap is used, there is no change of deviatoric plastic strains, while if the elliptical cap is used, the deviatoric components of the plastic strain change during the cap yielding. Fig. 3.9-4 shows the relation between the plane cap position ${}^t J_1^a$ and the volumetric plastic strain

$${}^t e_V^P = {}^t e_{11}^P + {}^t e_{22}^P + {}^t e_{33}^P$$

that is, the constraint that exists between these two quantities. We note that the plane cap movement corresponds to hardening

and that the volumetric plastic strain is constrained to be smaller than the input parameter W in absolute value.

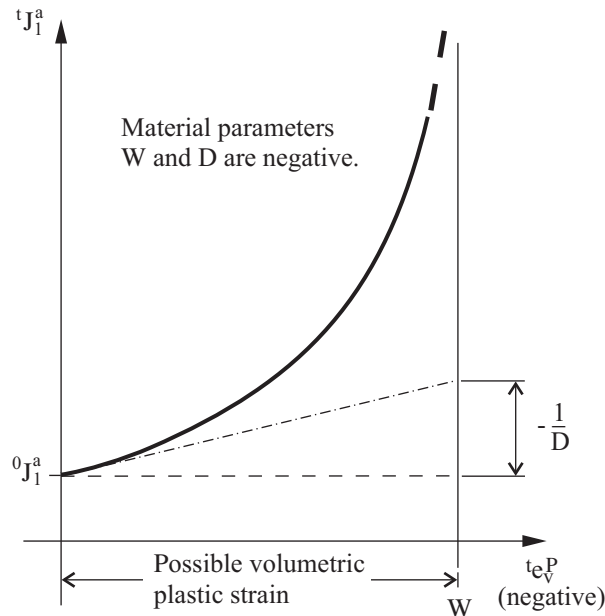


Figure 3.9-4: Relation between plane cap position tJ_1^a and volumetric plastic strain te_v^P

- In the case of vertex yielding (the stress state is represented by point H in Fig. 3.9-3), the plastic deformation corresponds to the Drucker-Prager and cap yielding. The vertex yielding leads to changes in volumetric and deviatoric plastic strains with the cap hardening behavior.
- Stress states beyond the tension cut-off value T (input to ADINA as a positive value) are not possible. When this limit is reached or exceeded, the program sets all the shear stress components equal to zero and the normal stress components all equal to $\frac{T}{3}$. The elastic constitutive law is used for the stiffness matrix.

3.9.3 Cam-clay material model

- The Cam-clay material model is a pressure-dependent plasticity model. It is based on
 - ▶ An associated flow rule using an elliptical yield function
 - ▶ The critical state line, which controls the failure of the material
 - ▶ The consolidation behavior of clayey materials
- The Cam-clay model can be used with the **2-D solid** and **3-D solid** elements.
- The Cam-clay model can be used with the **small displacement/small strain** and **large displacement/small strain** formulations.

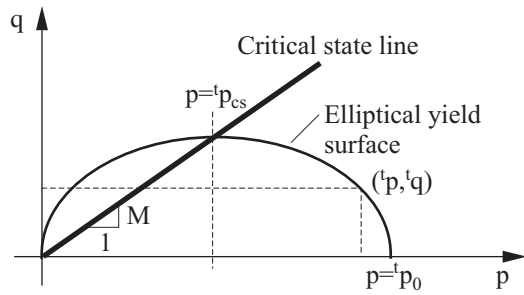
When used with the small displacement formulation, a materially-nonlinear-only formulation is employed and when used with the large displacement/small strain formulation, a TL formulation is employed.

- The Cam-clay model is able to simulate the following mechanical behaviors for clayey materials, which are confirmed by lab tests and in-situ tests:
 - ▶ Strain hardening and softening under normal consolidation states or overconsolidation states
 - ▶ Nonlinear dependence of the elastic volumetric strain on the hydrostatic pressure
 - ▶ An ultimate condition of perfect plasticity at which plastic shearing can continue indefinitely without changes in volume or effective stress
- The Cam-clay yield function is given by

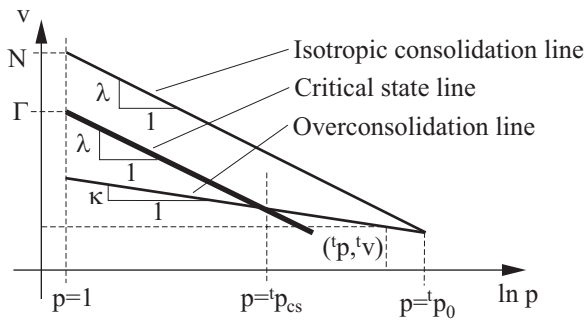
$${}^tF = \frac{{}^tq^2}{M^2} + {}^tp \left({}^tp - {}^tp_0 \right) = 0$$

where the mean stress ${}^t p$ and the distortional stress ${}^t q$ at time t are related to the first stress invariant ${}^t I_1$ and the second deviatoric stress invariant ${}^t J_2$ by ${}^t p = \frac{{}^t I_1}{3}$ and ${}^t q = \sqrt{3 {}^t J_2}$. M , a material constant, is the slope of the critical state line and ${}^t p_0$, called the pre-consolidation pressure, is the diameter of the ellipsoid at time t along the axis p . All of the variables are shown in Fig. 3.9-5(a).

- The hardening rule is written as



a) Cam-clay yield surface



b) Compression behavior

Figure 3.9-5: Cam-clay model

$${}^t v = N - \lambda \ln {}^t p_0 + \kappa \ln \frac{{}^t p_0}{{}^t p}$$

where ${}^t v$ is the specific volume at time t and ${}^t \dot{v}^P$ is the plastic specific volume rate at time t . λ , κ and Γ are material constants, in which λ is the slope of the isotropic consolidation line, κ is the slope of the overconsolidation line and Γ is the specific volume at the critical state when ${}^t p$ is 1.0. Γ is related to N , the specific volume at the isotropic consolidation state when ${}^t p$ is 1.0, by

$$\Gamma = N + (\kappa - \lambda) \ln p_0$$

- The effective bulk modulus at time t can be expressed as

$${}^t K = \frac{{}^t v {}^t p}{\kappa}$$

The corresponding shear modulus at time t is obtained using

$${}^t G = \frac{3(1-2\nu)}{2(1+\nu)} {}^t K, \text{ in which } \nu \text{ is the Poisson's ratio (constant}$$

throughout the analysis).

- In an analysis using the Cam-clay model, ADINA requires either the initial stresses or an initial stiffness to be defined.

Initial stresses: The initial stresses are either directly input or elastically calculated before or at the first load step for any specified element group. They can be tiny isotropic compressive stresses (such as 5% of the gravity-induced stresses), or can be the full gravity-induced stresses.

When the analysis starts with initial stresses, the initial size of the yield surface can either be directly input or computed from the initial stresses. If the initial stresses are induced by the full gravity load, the initial size of the yield surface (pre-consolidation pressure) would be estimated by employing the input parameters OCR and KNULL, unless the initial size of the

yield surface is specified in the Cam-clay model definition.

If the initial stresses are not directly input, ADINA calculates the initial stresses as the stresses computed at load step 1. Then,

- (a) The results for load step 1 are calculated using the input elastic material properties E and ν .
- (b) The results for the remaining load steps depend upon the solution for load step 1.

When the analysis starts with initial stresses, ADINA takes the true initial size of the yield surface as the greater of the specified initial size and the computed initial size.

Initial stiffness: The analysis can be performed with an initial stiffness, which is calculated using the initial Young's modulus and Poisson's ratio. In this case, the initial size of the yield surface must be directly specified.

- Parameter KNULL is defined as the coefficient of earth pressure at the normal consolidation state. Usually it is different than the coefficient of current earth pressure. KNULL is approximated by $K_0 = 1 - \sin \phi$ where ϕ is the internal friction angle of the soil.

- The material constant M is computed based on ϕ . If triaxial compression tests are performed, M is given by $M = \frac{6 \sin \phi}{3 - \sin \phi}$. If triaxial extension tests are performed, M is given by $M = \frac{6 \sin \phi}{3 + \sin \phi}$.

- For more information, see references:

ref. D.M. Wood, *Soil behavior and critical state soil mechanics*, Cambridge University Press, pp. 84-225, 1990.

ref. A.M. Britto and M.J. Gunn, *Critical state soil mechanics via finite elements*, Ellis Horwood Limited, pp. 161-184, 1987.

3.9.4 Mohr-Coulomb material model

- The Mohr-Coulomb model is based on
 - ▶ a linearly elastic and perfectly plastic yield behavior;
 - ▶ a yield function defined by the Mohr-Coulomb criterion;
 - ▶ a potential function defined by the Drucker-Prager criterion;
 - ▶ a piecewise tension cut-off analysis; and
 - ▶ a non-associative flow rule.
- The Mohr-Coulomb model can be used with the **2-D solid** and **3-D solid** elements.
- The Mohr-Coulomb model can be used with the **small displacement/small strain** and **large displacement/small strain** formulations.

When used with the small displacement/small strain formulation, a materially-nonlinear-only (MNO) formulation is employed, and when used with the large displacement/small strain formulation, a total Lagrangian (TL) formulation is employed.

- The Mohr-Coulomb yield function is given by

$${}^t f_{MC} = \sin \phi {}^t I_1 + \frac{1}{2} \left(3(1 - \sin \phi) \sin {}^t \theta + \sqrt{3} (3 + \sin \phi) \cos {}^t \theta \right) \sqrt{{}^t J_2} - 3c \cos \phi$$

where

$${}^t \theta = \frac{1}{3} \arccos \left(\frac{3\sqrt{3} {}^t J_3}{2({}^t J_2)^{3/2}} \right),$$

is the angle of similarity or Lode angle defined in the deviatoric stress plane (see Fig. 3.9-8), and $0 \leq {}^t \theta \leq \frac{\pi}{3}$. In addition, note that

- ▶ ϕ is the friction angle (a material parameter) such that

$$0 \leq \phi < \frac{\pi}{2};$$
- ▶ c is the cohesion (a material parameter) such that $c \geq 0$;
- ▶ tI_1 is the first stress invariant at time t ;
- ▶ tJ_2 is the second deviatoric stress invariant at time t ; and
- ▶ tJ_3 is the third deviatoric stress invariant at time t .

In the stress space, the Mohr-Coulomb yield function is an irregular hexagonal pyramid with its apex on the hydrostatic axis (see Fig. 3.9-6).

- The Mohr-Coulomb potential function is defined by the Drucker-Prager criterion such that its surface is *inscribed* in the Mohr-Coulomb surface defined by the dilatation angle ψ (a material parameter), where $0 \leq \psi \leq \phi$ (see Figs. 3.9-7 and 3.9-10). Consequently, the Mohr-Coulomb potential function is given by

$${}^tg_{MC} = \alpha_{MC} {}^tI_1 + \sqrt{{}^tJ_2} - k_{MC}$$

where

$$\alpha_{MC} = \frac{2 \sin \psi}{3(1 - \sin \psi) \sin \theta_{MC} + \sqrt{3}(3 + \sin \psi) \cos \theta_{MC}}.$$

The angle θ_{MC} is determined by inscribing a Drucker-Prager surface to the Mohr-Coulomb surface defined by the dilatation angle (see Fig. 3.9-10 assuming that $\psi = \phi$) and k_{MC} is not needed by the model. It is important to mention that this is the *default* Mohr-Coulomb potential function. Moreover, this function provides a plastic return to the yield surface that is unique and radial in the deviatoric stress plane. However, in the meridional

plane the plastic return is not unique at the apex of the cone or in the tension cut-off region.

- The tension cut-off (T) defines a limiting surface in the stress space via

$$I_1 - T = 0,$$

where $T \geq 0$ (see Fig. 3.9-7). The tension cut-off (a material parameter) can be taken to be three times the tensile strength of the material. This limiting surface combined with the Mohr-Coulomb yield surface can better approximate the behavior of a material when tensile stresses occur and completely define ADINA's Mohr-Coulomb yield surface. Note that when a tension cut-off analysis is applied by the model, the hydrostatic stress components are shifted to the hydrostatic pressure corresponding to the tension cut-off value.

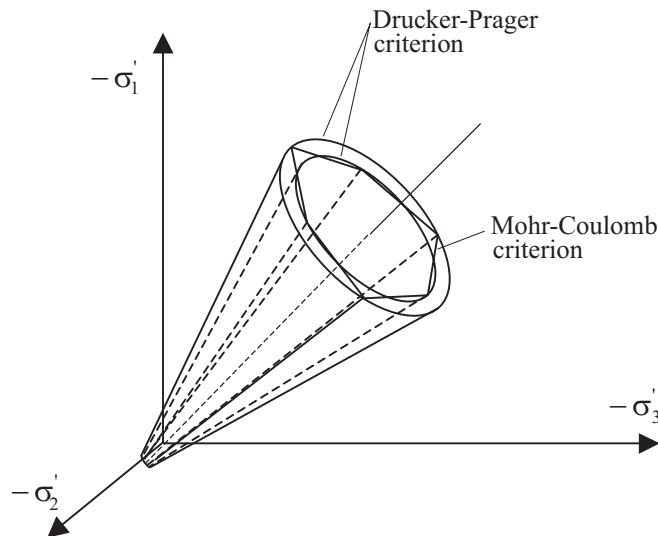


Figure 3.9-6: The yield surfaces of the Mohr-Coulomb model and the Drucker-Prager model in stress space

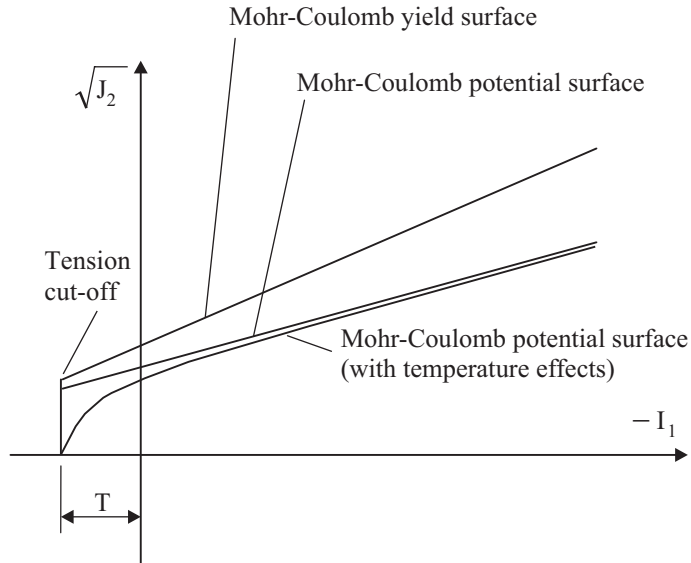


Figure 3.9-7: The Mohr-Coulomb model in the I_1 vs $\sqrt{J_2}$ plane (or meridional plane)

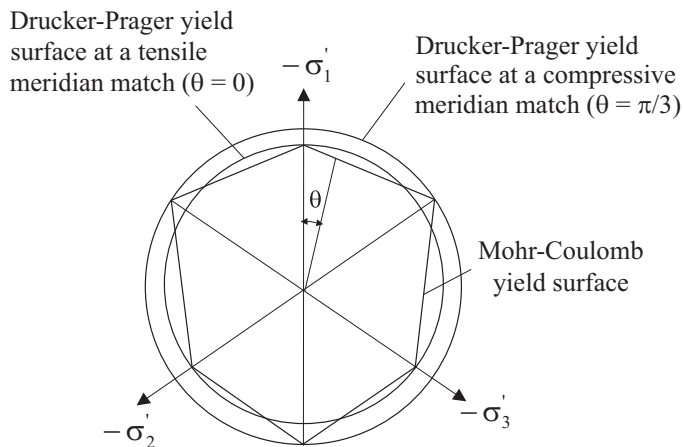


Figure 3.9-8: Shapes of the Mohr-Coulomb and the Drucker-Prager yield criteria in the π plane

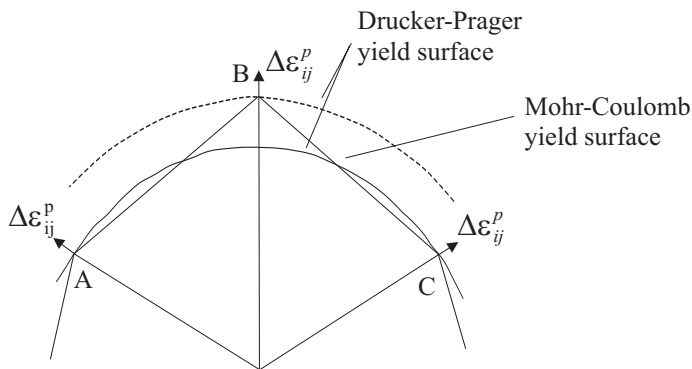


Figure 3.9-9: Corner treatment of the Mohr-Coulomb model

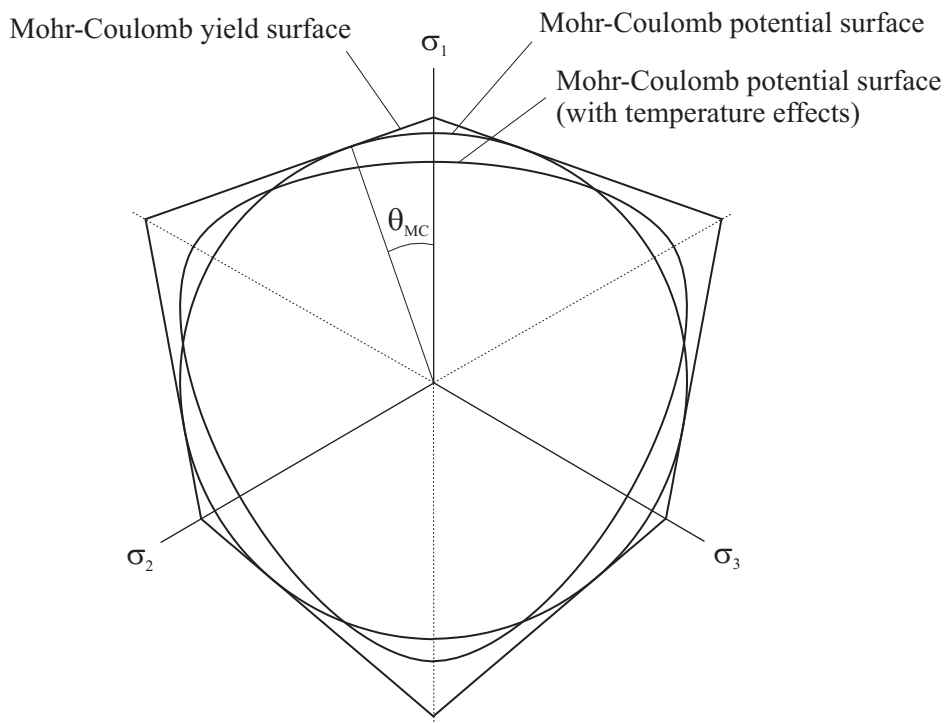


Figure 3.9-10: Shapes of the Mohr-Coulomb yield and potential surfaces in the π plane

- In the limiting cases of vertex yielding, that is, the stress states represented by A or C (tensile meridian, $\theta = 0$) and B (compressive meridian, $\theta = \frac{\pi}{3}$) in Fig. 3.9-9, the analysis boils down to a Drucker-Prager-type analysis.
- A couple of important and practical suggestions apply to the Mohr-Coulomb material model.
 - ▶ Since the model follows a non-associative flow rule, the constitutive matrices at the integration points are, in general, non-symmetric. Therefore, it is recommended to use the non-symmetric sparse solver instead of the default sparse solver when a user's model exhibits slow convergence or it cannot converge (see the AUI command MASTER parameter SOLVER=NONSYM-SP).
 - ▶ If tensile conditions are present in a user's model, then it is recommended to use the potential function implemented for the Mohr-Coulomb material model with temperature effects (see below). In order to successfully use this potential function, some type of temperature input to ADINA is required in the user's model; however, these can be set to zero so as to neglect the temperature effects.
- For more information, see the following references:
 - ref. W. F. Chen and E. Mizuno, *Nonlinear Analysis in Soil Mechanics, Theory and Implementation, Developments in Geotechnical Engineering 53*, Elsevier, Amsterdam, 1990.
 - ref. M. A. Crisfield, *Non-linear Finite Element Analysis of Solids and Structures, Volume 2: Advanced Topics*, John Wiley & Sons, Chichester, 1997.

3.9.4.1 Mohr-Coulomb material model with temperature effects

- The Mohr-Coulomb material model with temperature effects has basically the same characteristics of the default Mohr-Coulomb material model except for the points discussed below.
- In some geotechnical analyses, the thermal strains ${}^t e^{TH}$ need to be included. These strains are taken into account by assuming that the Mohr-Coulomb material is an isotropic temperature independent material (see Section 3.1.4) affecting the material behavior only hydrostatically. The thermal strains are calculated from the temperature conditions and α the mean coefficient of thermal expansion, where $\alpha \geq 0$, as given in Eq. (3.1-16).
- ADINA will require temperature data as input. Please refer to Section 5.6 for a discussion on the several ways to input these data to ADINA.
- The Mohr-Coulomb potential function used describes a completely smooth surface in the stress space, as shown in Figs. 3.9-7 and 3.9-10, and it is given by

$${}^t g_{MC} = \sqrt{(2a_{MC} \sin \psi)^2 + \left({}^t r M_{MC} \sqrt{{}^t J_2}\right)^2} + 2 \sin \psi \left({}^t I_1 - \frac{3c}{\tan \phi}\right),$$

where

$$a_{MC} = \begin{cases} \frac{3c\varepsilon}{\tan \phi} & \text{if } T \geq \frac{3c}{\tan \phi} \quad \text{with } 0 < \varepsilon < 1, \\ \frac{3c}{\tan \phi} - T & \text{if } T < \frac{3c}{\tan \phi}, \end{cases}$$

$$M_{MC} = \sqrt{3}(3 - \sin \phi)$$

and

$${}^t r = \frac{4(1 - e^2) \cos^2 {}^t \theta + (2e - 1)^2}{2(1 - e^2) \cos {}^t \theta + (2e - 1) \sqrt{4(1 - e^2) \cos^2 {}^t \theta + 5e^2 - 4e}},$$

is the elliptic function described in

- ref. Ph. Menétrey and K. J. Willam, “Triaxial Failure Criterion for Concrete and Its Generalization”, *ACI Structural Journal*, 92(3), pp. 311-318, 1995.

and is defined for $0 \leq \theta \leq \frac{\pi}{3}$. In order for such an elliptic function to be convex and smooth, it is required that $0.5 < e \leq 1$, where e is known as the eccentricity (in the deviatoric plane) and it will be assumed to be

$$e = \frac{3 - \sin \phi}{3 + \sin \phi},$$

as suggested in the above reference. Basically, this ensures that the surface described by the above potential function matches the Mohr-Coulomb yield surface at the tensile meridian ($\theta = 0$) and compressive meridian ($\theta = \frac{\pi}{3}$). This potential function contains parameters and terms that are mentioned in the description of the default Mohr-Coulomb material model. Note that the accepted value ranges of some of these quantities have changed due to the nature of the new function. This potential function contains the following parameters and terms:

- ▶ c is the cohesion such that $c > 0$;
- ▶ ϕ is the friction angle such that $0 < \phi < \frac{\pi}{2}$;
- ▶ ψ is the dilatation angle such that $0 < \psi \leq \phi$;
- ▶ a_{MC} is the distance along the hydrostatic axis from the apex of the Mohr-Coulomb yield surface to the surface described by the potential function;

- ▶ ε is an eccentricity parameter in the meridional plane (at the apex of the Mohr-Coulomb yield surface) that helps define a_{MC} such that $0 < \varepsilon < 1$ and it is recommended to be $\varepsilon \approx 0.1$;
 - ▶ T is the tension cut-off such that $T \geq 0$; and
 - ▶ ${}^t\theta$ is the angle of similarity defined earlier and is a function of tJ_2 and tJ_3 .
- The Mohr-Coulomb potential function used provides a smooth surface in the stress space and, therefore, in both the deviatoric and meridional stress planes. In the former plane, it has an elliptic shape, which is basically dictated by the elliptic function (see Fig. 3.9.10), while in the latter plane its shape is that of a hyperbola (see Fig. 3.9.7). The flow rule is non-associative and presents a unique plastic return in the stress space, which is guaranteed by the smoothness of the potential surface. Note that all the above characteristics ensure the need for no special cases at the corners of the Mohr-Coulomb yield surface, its apex or in the tension cut-off region.
 - Due to the non-associative flow rule applied in the model, it is recommended to use the non-symmetric sparse solver instead of the default sparse solver when a user's model exhibits slow convergence or it cannot converge (see the AUI command MASTER parameter SOLVER=NONSYM-SP).

3.10 Fabric material model with wrinkling

- The elastic-orthotropic material model (discussed in Section 3.2) can be used with the **2-D solid element** with the 3-D plane stress (membrane) option and the **large displacement/small strain formulation** to model the behavior of fabric structures.
- Fabric materials are typically very thin and flexible. They can support only in-plane tensile stresses, and will wrinkle under compressive stresses.

- The analysis of fabric structures with ADINA is performed in two steps: first, initial strains must be specified, Young's moduli must be set to a fraction (usually 10^{-4}) of their actual values, and the wrinkling of the material must not be taken into account. Then, in a second step (restart analysis), the same orthotropic material model is used except that the Young's moduli are now set to their actual values, and wrinkling is taken into account.

- For each integration point, wrinkling occurs if the smallest of the two in-plane principal stresses $({}^t\sigma_{p1}, {}^t\sigma_{p2})$ becomes negative.

If wrinkling occurs, then this smallest principal stress ${}^t\sigma_{p2}$ is set to zero. If the other in-plane principal stress ${}^t\sigma_{p1}$ is greater than zero, the program calculates a new "wrinkling stress" ${}^t\sigma'_{p1}$ using

$${}^t\sigma'_{p1} = A' {}^t\varepsilon_{p1}$$

where ${}^t\varepsilon_{p1}$ is the strain in the direction of ${}^t\sigma_{p1}$ and A is calculated from the orthotropic material law, using the condition ${}^t\sigma_{p2} = 0$. If ${}^t\sigma_{p1}$ is smaller than zero, ADINA sets ${}^t\sigma_{p1}$ to zero; in this case, wrinkling occurs in two directions.

The element constitutive matrix corresponding to the principal directions is also modified for the wrinkling behavior and becomes

$$\begin{bmatrix} \alpha A & 0 & 0 \\ 0 & \eta A & 0 \\ 0 & 0 & \eta A \end{bmatrix}$$

where η is a reduction factor equal to 10^{-5} , and $\alpha = 1$ if ${}^t\sigma_{p1} \geq 0$ or $\alpha = \eta$ if ${}^t\sigma_{p1} < 0$.

3.11 Viscoelastic material model

- The viscoelastic model can be used with the **truss**, **2-D solid**, **3-D solid** and **shell** elements.

- The viscoelastic model can be used with the **small displacement/small strain, large displacement/small strain** and **large displacement/large strain** formulations (2-D solid and 3-D solid elements only).

When used with the small displacement/small strain formulation, a materially-nonlinear-only formulation is employed, when used with the large displacement/small strain formulation, a TL formulation is employed and when used with the large displacement/large strain formulation, the ULH formulation is employed.

- The mechanical behavior for an isotropic and linear viscoelastic material may be expressed in tensor notation as

$$s_{ij}(t) = 2G(0)e_{ij}(t) + 2 \int_{0^+}^t e_{ij}(t-\tau) \frac{dG(\tau)}{d\tau} d\tau \quad (3.11-1)$$

$$\sigma_{kk}(t) = 3K(0)\varepsilon_{kk}(t) + 3 \int_{0^+}^t \varepsilon_{kk}(t-\tau) \frac{dK(\tau)}{d\tau} d\tau \quad (3.11-2)$$

where t is the time, $s_{ij} = \sigma_{ij} - \frac{1}{3}\delta_{ij}\sigma_{kk}$ is the deviatoric stress, δ_{ij} is the Kronecker delta, σ_{ij} is the stress, $e_{ij} = \varepsilon_{ij} - \frac{1}{3}\delta_{ij}\varepsilon_{kk}$ is the deviatoric strain, ε_{ij} is the strain, $G(t)$ is the shear modulus and $K(t)$ is the bulk modulus.

In the presence of a temperature variation $\theta(t)$, the stresses for an isotropic and thermorheologically linear viscoelastic material may be written as

$$s_{ij}(t) = 2G(0)e_{ij}(t) + 2 \int_{0^+}^{\xi} e_{ij}(\xi-\zeta) \frac{dG(\zeta)}{d\zeta} d\zeta \quad (3.11-3)$$

$$\sigma_{kk}(t) = 3K(0)(\varepsilon_{kk}(t) - \varepsilon_{kk}^{th}(t)) + 3 \int_{0^+}^{\xi} [\varepsilon_{kk}(\xi-\zeta)] \frac{dK(\zeta)}{d\zeta} d\zeta - 3\varepsilon_{kk}^{th}(t) \int_{0^+}^{\xi} \frac{dK(\zeta)}{d\zeta} d\zeta$$

(3.11-4)

where

$$\zeta = \int_0^t \psi[\theta(\eta)] d\eta, \quad \xi = \int_0^t \psi[\theta(\tau)] d\tau \quad (3.11-5)$$

and the thermal strain is given by

$$\varepsilon_{kk}^{th}(t) = 3\alpha(\theta(t))[\theta(t) - \theta_{TALPHA}] - 3\alpha(\theta_0)[\theta_0 - \theta_{TALPHA}] \quad (3.11-6)$$

$\alpha(\theta(t))$ is the temperature-dependent coefficient of thermal expansion and $\psi(t)$ is the shift function, which obeys

$$\psi(\theta_{TALPHA}) = 1, \quad \psi(\theta) > 0, \quad \frac{d\psi}{d\theta} > 0 \quad (3.11-7)$$

Note that θ_{TALPHA} is the reference temperature used for thermal strain calculation.

In equations (3.11-3) and (3.11-4) it is assumed that the mechanical and thermal responses are uncoupled. Furthermore if the temperature is constant, equations (3.11-3) and (3.11-4) reduce to equations (3.11-1) and (3.11-2).

- We assume the following thermo-material properties:

$$G(t) = G_\infty + \sum_{i=1}^{\eta_G} G_i e^{-\beta_i t} \quad (3.11-8)$$

$$K(t) = K_\infty + \sum_{i=1}^{\eta_K} K_i e^{-\gamma_i t} \quad (3.11-9)$$

$$\theta(t) \neq 0 \quad (3.11-10)$$

$$\alpha = \alpha(\theta(t)) \quad (3.11-11)$$

where G_∞ and K_∞ are the long-time shear modulus and bulk modulus respectively, β_i and γ_i are the decay constants for the

shear modulus and bulk modulus respectively and η_G and η_K are the number of time-dependent terms for the shear modulus and bulk modulus respectively. Equations (3.11-8) and (3.11-9) are referred to in the literature as Prony or Dirichlet series. η_G and η_K are limited to a maximum value of 15.

The shift function used is either the Williams-Landell-Ferry (WLF) equation, written as follows

$$\log_{10} \psi(\theta) = -\frac{C_1(\theta - \theta_{ref})}{C_2 + (\theta - \theta_{ref})} \quad (3.11-12)$$

or the Arrhenius shift function

$$\begin{aligned} \log_{10} \psi(\theta) &= C_1 \left(\frac{1}{\theta} - \frac{1}{\theta_{ref}} \right), \theta \geq \theta_{ref} \\ &= C_2 \left(\frac{1}{\theta} - \frac{1}{\theta_{ref}} \right), \theta < \theta_{ref} \end{aligned} \quad (3.11-13)$$

in which C_1 and C_2 are material constants, and θ_{ref} is defined as the reference temperature of the WLF equation (TREF in the the MATERIAL VISCOELASTIC command).

- You can either enter the Prony series constants in the AUI, or you can enter the measured data for the shear and bulk modulus and let the AUI construct the Prony series. In the latter case, the AUI constructs the Prony series using a least-squares technique, see the reference by Frutiger and Woo given below.
- The nodal point temperatures are input to ADINA as discussed in Section 5.6.
- For more information, see the following references:
 - ref. W.N. Findley, J.S. Lai and K. Onaran, *Creep and relaxation of nonlinear viscoelastic materials*, Dover Publications, 1976.
 - ref. R.L. Frutiger and T.C. Woo, "A thermoviscoelastic

analysis for circular plates of thermorheologically simple material”, *Journal of Thermal Stresses*, 2:45-60, 1979.

3.12 Porous media formulation

- The porous media formulation is applicable to porous structures subject to static or dynamic loading. It deals with the interaction between the porous solids and pore fluids, which flow through the porous solid skeleton.
- The porous media formulation only applies to 2D and 3D solid elements. These elements have pore pressure variables at their corner node points.
- The pore pressures are considered to be independent variables, along with the displacements. The pore pressures are output by ADINA along with the displacements, velocities, stresses and other solution results.
- The porous media model is implemented with a fully coupled formulation, as described in detail below.
- The ADINA porous media formulation is based on the following assumptions:
 - ▶ The pore fluids are incompressible or slightly compressible.
 - ▶ The porous solid skeleton is fully saturated with pore fluid.
 - ▶ The pore fluid flows through the porous solid according to Darcy’s law.
- With the porous media formulation, undrained analysis (static pore fluid) and consolidation or swelling analysis (pore fluid flowing through the porous solid skeleton) can be performed.
- In the analyses of porous structures, all the nonlinear formulations in ADINA (small displacement/small strain, large displacement/small strain and large displacement/large strain) can be used. A materially-nonlinear-only formulation is employed for the small displacement/small strain formulation; the TL formulation for the large displacement/small strain formulation and

the ULH formulation for the large displacement/large strain formulation.

- The mixed (u/p) formulation can be employed in the analyses of porous structures. However the incompatible modes feature cannot be employed in the analyses of porous structures.
- The types of problems for which the porous media model can be employed are:

Transient static analysis (consolidation): The pore pressure distribution, the displacement and stress distribution, and their changes with time are of great interest.

Transient dynamic analysis: In addition to the pore pressure, displacement and stress results, the potential failure of the porous structure will draw much more attention. Liquefaction, which is an extremely important failure mode for infrastructures in an earthquake disaster, can be studied with the porous media model.

Undrained analysis: No specific formulation is given to relate the pore pressure to total stresses under undrained conditions. The undrained analysis can be performed by specifying undrained boundary conditions and small time steps.

- The orthotropic, thermo-isotropic, thermo-orthotropic, thermo-plastic, Drucker-Prager, Mohr-Coulomb, Cam-clay, creep, and plastic-creep material models can be employed in porous media analysis.
- The general equations governing the porous media model are as follows:

- ▶ The macroscopic stresses must satisfy the equilibrium condition:

$$\sigma_{ij,j} + f_i = 0 \quad (3.12-1)$$

where the total (Cauchy) stress is

$$\sigma_{ij} = C_{ijkl}^e \varepsilon_{kl}^e + P_f \delta_{ij}, \quad i, j, k, l = 1, 2, 3 \quad (3.12-2)$$

P_f is the pore pressure, C_{ijkl}^e is the fourth order elasticity tensor, ε_{kl}^e is the elastic strain tensor and δ_{ij} is the Kronecker delta.

► The continuity condition for slightly compressible fluid flows must be satisfied:

$$\nabla \cdot (\mathbf{k} \cdot \nabla P_f) = \frac{\partial \varepsilon_{ii}}{\partial t} + \frac{n}{\kappa_f} \frac{\partial P_f}{\partial t} \quad (3.12-3)$$

where \mathbf{k} is the permeability matrix of the porous material, ε_{ii} is the volumetric strain of the porous skeleton, n is the porosity of the material and κ_f is the bulk modulus of the fluid. If the fluid is assumed incompressible, the last term in (3.12-3) will be zero.

- After a finite element discretization, equations (3.12-1) and (3.12-3) yield the following equations.

Transient statics:

$$\begin{aligned} \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ {}^{t+\Delta t} \mathbf{K}_{u \ p_f}^T & {}^{t+\Delta t} \mathbf{K}_{p_f \ p_f}^{(1)} \end{bmatrix} \begin{bmatrix} {}^{t+\Delta t} \dot{\mathbf{U}} \\ {}^{t+\Delta t} \dot{\mathbf{P}}_f \end{bmatrix} + \begin{bmatrix} {}^{t+\Delta t} \mathbf{K}_{uu} & {}^{t+\Delta t} \mathbf{K}_{u \ p_f} \\ \mathbf{0} & -{}^{t+\Delta t} \mathbf{K}_{p_f \ p_f}^{(2)} \end{bmatrix} \begin{bmatrix} {}^{t+\Delta t} \mathbf{U} \\ {}^{t+\Delta t} \mathbf{P}_f \end{bmatrix} \\ = \begin{bmatrix} {}^{t+\Delta t} \mathbf{R}_u \\ {}^{t+\Delta t} \mathbf{R}_{p_f} \end{bmatrix} \end{aligned} \quad (3.12-4)$$

Transient dynamics:

$$\begin{aligned}
& \begin{bmatrix} {}^{t+\Delta t} \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} {}^{t+\Delta t} \ddot{\mathbf{U}} \\ {}^{t+\Delta t} \ddot{\mathbf{P}}_f \end{bmatrix} + \begin{bmatrix} \mathbf{C} & \mathbf{0} \\ {}^{t+\Delta t} \mathbf{K}_{u p_f}^T & {}^{t+\Delta t} \mathbf{K}_{p_f p_f}^{(1)} \end{bmatrix} \begin{bmatrix} {}^{t+\Delta t} \dot{\mathbf{U}} \\ {}^{t+\Delta t} \dot{\mathbf{P}}_f \end{bmatrix} + \\
& \begin{bmatrix} {}^{t+\Delta t} \mathbf{K}_{uu} & {}^{t+\Delta t} \mathbf{K}_{u p_f} \\ \mathbf{0} & -{}^{t+\Delta t} \mathbf{K}_{p_f p_f}^{(2)} \end{bmatrix} \begin{bmatrix} {}^{t+\Delta t} \mathbf{U} \\ {}^{t+\Delta t} \mathbf{P}_f \end{bmatrix} = \begin{bmatrix} {}^{t+\Delta t} \mathbf{R}_u \\ {}^{t+\Delta t} \mathbf{R}_{p_f} \end{bmatrix} \quad (3.12-5)
\end{aligned}$$

For a finite element, these matrices are defined as follows (omitting the left superscript for brevity):

$$\begin{aligned}
\mathbf{K}_{uu} &= \int_V \mathbf{B}_u^T \mathbf{D} \mathbf{B}_u dV \\
\mathbf{K}_{u p_f} &= \int_V \mathbf{B}_u^T \mathbf{I} \mathbf{H}_{p_f} dV \\
\mathbf{K}_{p_f p_f}^{(1)} &= \frac{1}{\kappa_f} \int_V n \mathbf{H}_{p_f}^T \mathbf{H}_{p_f} dV \\
\mathbf{K}_{p_f p_f}^{(2)} &= \int_V \mathbf{B}_{p_f}^T \mathbf{k} \mathbf{B}_{p_f} dV \\
\mathbf{R}_u &= \int_V \mathbf{H}_u^T \mathbf{f} dV + \int_{S_f} \left(\mathbf{H}_u^s \right)^T \mathbf{f} dS_f \\
\mathbf{R}_{p_f} &= \int_{S_q} \left(\mathbf{H}_{p_f}^s \right)^T \mathbf{q} dS_q
\end{aligned}$$

In the above expressions, \mathbf{D} is the constitutive matrix (possibly elasto-plastic), \mathbf{f} and \mathbf{q} are load vectors (force, traction and flux), \mathbf{B}_u and \mathbf{B}_{p_f} are shape function matrices for the displacements \mathbf{U} and pore pressures \mathbf{P}_f , \mathbf{H}_u and \mathbf{H}_{p_f} are shape function matrices for the displacements \mathbf{U} and pore pressures \mathbf{P}_f , and \mathbf{I} is the identity tensor.

- Numerically integrating equations (3.12-4) and (3.12-5) with respect to time gives the final equations used for the porous media model:

Transient statics: the Euler backward method is used:

$$\begin{aligned}
 & \begin{bmatrix} {}^{t+\Delta t}\mathbf{K}_{uu} & {}^{t+\Delta t}\mathbf{K}_{u p_f} \\ {}^{t+\Delta t}\mathbf{K}_{u p_f}^T & {}^{t+\Delta t}\mathbf{K}_{p_f p_f}^{(1)} - \Delta t {}^{t+\Delta t}\mathbf{K}_{p_f p_f}^{(2)} \end{bmatrix} \begin{bmatrix} {}^{t+\Delta t}\mathbf{U} \\ {}^{t+\Delta t}\mathbf{P}_f \end{bmatrix} \\
 &= \begin{bmatrix} {}^{t+\Delta t}\mathbf{R}_u \\ -\Delta t {}^{t+\Delta t}\mathbf{R}_{p_f} + {}^{t+\Delta t}\mathbf{K}_{u p_f}^T {}^t\mathbf{U} + {}^{t+\Delta t}\mathbf{K}_{p_f p_f}^{(1)} {}^t\mathbf{P}_f \end{bmatrix}
 \end{aligned} \quad (3.12-6)$$

Transient dynamics: the Euler backward method is used for the continuity equation and the Newmark method is used for the equilibrium equation:

$$\begin{aligned}
 & \begin{bmatrix} {}^{t+\Delta t}\mathbf{K}_{uu} + a_0 {}^{t+\Delta t}\mathbf{M} + a_1 {}^{t+\Delta t}\mathbf{C} & {}^{t+\Delta t}\mathbf{K}_{u p_f} \\ {}^{t+\Delta t}\mathbf{K}_{u p_f}^T & {}^{t+\Delta t}\mathbf{K}_{p_f p_f}^{(1)} - \Delta t {}^{t+\Delta t}\mathbf{K}_{p_f p_f}^{(2)} \end{bmatrix} \begin{bmatrix} {}^{t+\Delta t}\mathbf{U} \\ {}^{t+\Delta t}\mathbf{P}_f \end{bmatrix} \\
 &= \begin{bmatrix} {}^{t+\Delta t}\mathbf{R}_u^d \\ -\Delta t {}^{t+\Delta t}\mathbf{R}_{p_f} + {}^{t+\Delta t}\mathbf{K}_{u p_f}^T {}^t\mathbf{U} + {}^{t+\Delta t}\mathbf{K}_{p_f p_f}^{(1)} {}^t\mathbf{P}_f \end{bmatrix}
 \end{aligned} \quad (3.12-7)$$

in which

$$\begin{aligned}
 {}^{t+\Delta t}\mathbf{R}_u^d &= {}^{t+\Delta t}\mathbf{R}_u + {}^{t+\Delta t}\mathbf{M} \left(a_0 {}^t\mathbf{U} + a_2 {}^t\dot{\mathbf{U}} + a_3 {}^t\ddot{\mathbf{U}} \right) + \\
 & {}^{t+\Delta t}\mathbf{C} \left(a_1 {}^t\mathbf{U} + a_4 {}^t\dot{\mathbf{U}} + a_5 {}^t\ddot{\mathbf{U}} \right)
 \end{aligned} \quad (3.12-8)$$

and where a_0, a_1, \dots, a_5 are the integration constants for the Newmark method (see Ref. KJB, Section 9.2.4).

- Please note the following:
 - Usually, the global stiffness matrix is no longer positive definite in porous media analysis. This is because the diagonal components of ${}^{t+\Delta t}\mathbf{K}_{p_f p_f}^{(1)} - \Delta t {}^{t+\Delta t}\mathbf{K}_{p_f p_f}^{(2)}$ in equations (3.12-6) and (3.12-7) are usually negative. Therefore, you should

request that ADINA continue execution after ADINA detects a non-positive definite stiffness matrix (see Section 7.1.1).

- ▶ It is important to have a reasonable scheme for choosing time steps. Theoretically, the solution process is unconditionally stable for any time step size because of the use of the Euler backward method. However, this does not necessarily imply that any time step size is permissible. A small time step size and large mesh size will result in the zigzag pore pressure distribution at the initial stage of pore fluid dissipation (Britto & Gunn, 1987). It is logical to use progressively larger time steps in the porous media analysis.
 - ▶ The pore pressures are only computed by ADINA at the corner nodes of solid elements. During postprocessing, the AUI interpolates the pore pressures from the corner nodes to the other nodes (e.g. mid-side nodes) in higher-order elements.
 - ▶ Once the porous solid material properties are specified, the corresponding pore fluid properties are automatically computed. Certainly, they can be redefined if necessary.
 - ▶ Porous media cannot be used in explicit dynamic analysis.
- For more information, see the following reference:

ref. Britto, A.M. and Gunn, M.J., *Critical state soil mechanics via finite elements*, Ellis Horwood Limited, 1987.

3.13 Gasket material model

- Gaskets are relatively thin components placed between two bodies/surfaces to create a sealing effect and prevent fluid leakage (see Figure 3.13-1). While most gaskets are flat, any arbitrary gasket geometry can be modeled in ADINA.

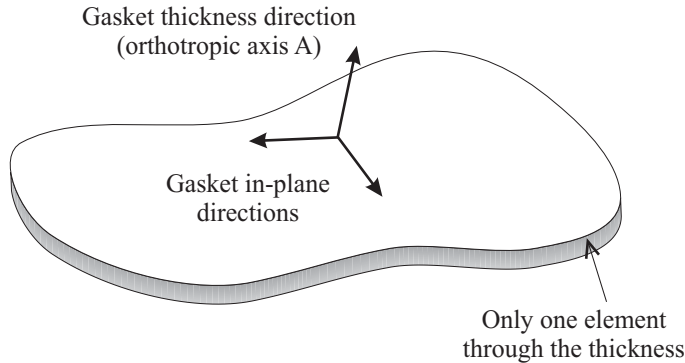


Figure 3.13-1: Schematic of gasket

- The sealing effect is created when the compressive load, applied in the direction of the gasket thickness, exceeds the initial yield stress of the gasket. The sealing effect is maintained as long as the compressive stress does not drop beyond a specified threshold value. The gasket ruptures if the compressive stress exceeds the gasket's ultimate stress. Unlike rupture, if a gasket leaks it still maintains its load-deflection characteristics.
- The gasket model can be used with 2-D plane strain, 2-D axisymmetric and 3-D solid elements. It can also be used with **small displacement/small strain, large displacement/small strain** formulations.
- The gasket behaves as a nonlinear elastic-plastic material when compressed in the thickness or gasket direction. Its load-deformation characteristics are represented by pressure-closure curves. Tensile stiffness can be assumed to be constant or zero. Hardening is assumed to be isotropic.
- Since the gasket material has different properties in different directions, it is considered by ADINA to be an orthotropic material. The gasket thickness direction is defined as the material axis a direction. The AUI attempts to determine this direction automatically, see below.
- The closure strain is always measured as the change in gasket thickness divided by the original gasket thickness.

- The gasket's uni-directional plasticity model speeds up computations, and allows more flexibility in defining the shape of the loading and unloading curves.
- Two gasket types are implemented in ADINA, the full or regular gasket, and the simple gasket.
- In the full gasket, elastic material properties are used for the in-plane deformations. Therefore the full gasket element has stiffness for all translational degrees of freedom.
- In the simple gasket, the element has no stiffness corresponding to the in-plane deformations. Therefore the nodal degrees of freedom for the in-plane directions must be eliminated, otherwise the model will have zero pivots in the stiffness matrix. The AUI attempts to automatically eliminate the nodal degrees of freedom for the in-plane directions, see below.

User input

The user input for the gasket material is now described in terms of the command-line interface.

```
MATERIAL GASKET NAME TREF DENSITY,  
                    YIELD-CURVE G-G E-G ALPHA-G,  
                    LEAKAGE-PRESSURE,  
                    E-INPLANE NU-INPLANE  
                    ALPHA-INPLANE ,NPOINTS
```

lcurve_i

E-G is the Young's modulus for tensile loading in the thickness direction ($E_{tensile}$). It is allowed to set E-G to a very small number, such as 1E-5, to model a gasket with very small tensile stiffness.

G-G, ALPHA-G are the transverse shear modulus and the coefficient of thermal expansion in the thickness direction.

LEAKAGE-PRESSURE specifies the leakage pressure. The LEAKAGE-PRESSURE does not affect the material response, but the LEAKAGE-PRESSURE affects the possible gasket states, see

below.

E-INPLANE, NU-INPLANE and ALPHA-INPLANE are the in-plane values of the Young's modulus, Poisson's ratio and coefficient of thermal expansion. These properties are ignored for a simple gasket.

YIELD-CURVE, NPOINTS and the lcurve_i are used to enter the pressure - closure strain relationship. YIELD-CURVE and the lcurve_i are curves that are defined using the LCURVE command.

Pressure - closure strain input

We now describe the pressure - closure strain input in more detail. Figure 3.13-2 shows a typical pressure - closure strain input. The main loading curve is specified by parameter YIELD-CURVE. The main loading curve can contain any number of points, and successive points are connected by straight lines.

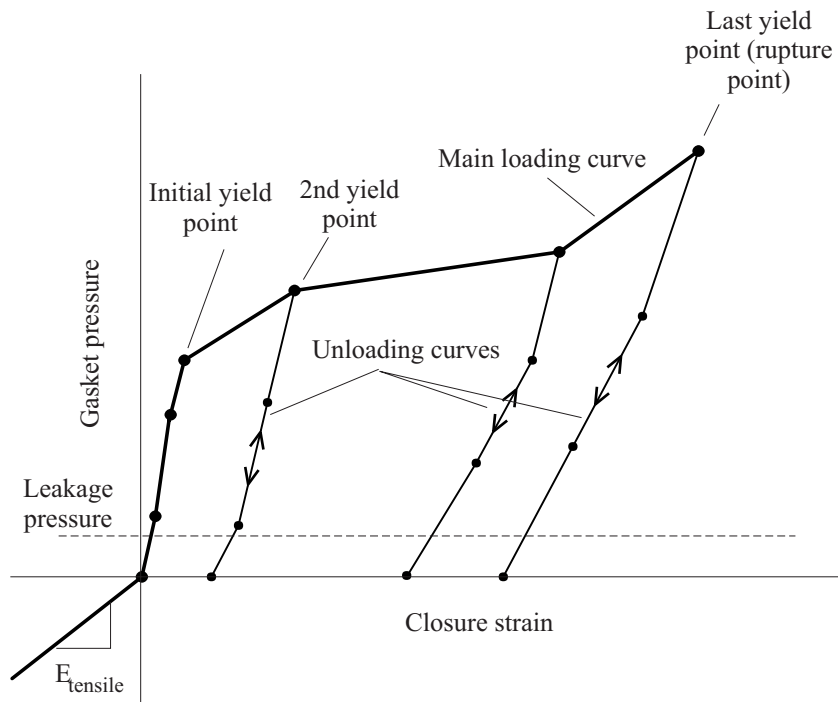


Figure 3.13-2: Pressure-closure strain input for a gasket material

The point number of the first yield point on the yield curve is given by parameter NPOINTS. NPOINTS must be greater than or equal to 3.

For each point on the main loading curve higher than the first yield point, there is an associated unloading curve, as shown. Notice that each unloading curve has the same number of points, and this number of points is equal to NPOINTS. There must be at least three unloading curves, and therefore the total number of points on the yield curve must be at least $\text{NPOINTS} + 2$.

There are two options for specification of each unloading curve:

- 1) The unloading curve can be specified using one of the lcurve_i data input lines. The lcurve must contain NPOINTS points, the first point must have zero pressure, and the last point must correspond to the (closure strain, pressure) of the main loading curve yield point.
- 2) The AUI can automatically construct the unloading curve by interpolation of the user-input unloading curves.

It is necessary to specify at least one of the unloading curves using the lcurve_i data input lines.

User input for nonlinear elastic gasket

We now describe how to input material data for a nonlinear elastic gasket.

Figure 3.13-3 shows an example for input of pressure - closure strain data for a nonlinear elastic gasket. In this example, we assume that points 2 to 4 correspond to experimental data. Points 5 to 7 are on a line extended from segment 3-4.

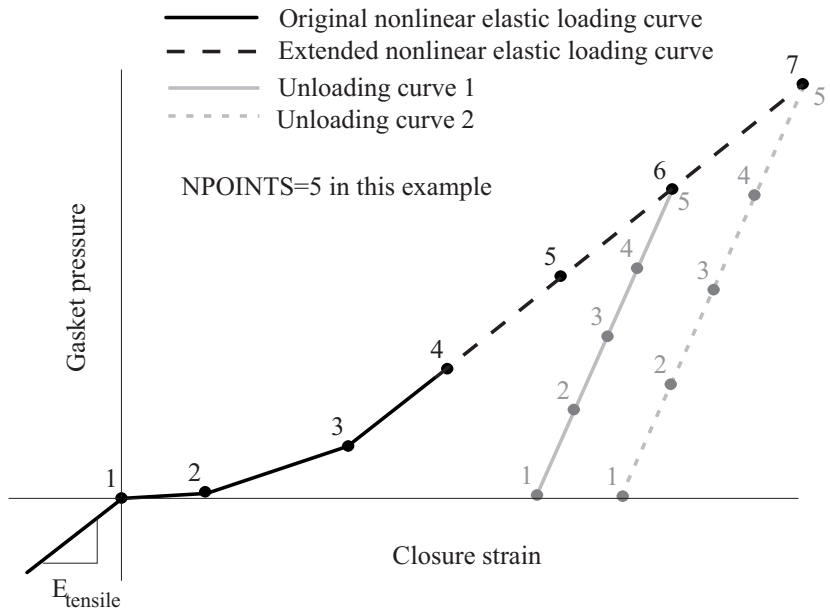


Figure 3.13-3: Example input for nonlinear elastic gasket

NPOINTS for this example is 5.

Points NPOINTS to NPOINTS+2 should correspond to very high closure strain, so that the program never reaches the closure strain corresponding to point NPOINTS.

The gasket material will be nonlinear elastic as long as the closure strain is less than the closure strain at point NPOINTS, and the relationship between closure strain and gasket pressure is given by the segments connecting points 1 to NPOINTS.

Two unloading curves must be entered, corresponding to points NPOINTS+1 and NPOINTS+2 on the loading curve, as shown. Each unloading curve has NPOINTS equally spaced points, with unloading point 1 for pressure =0 and unloading point NPOINTS for the point coincident with the loading curve.

The tensile behavior of the gasket is governed by constant E-G ($E_{tensile}$).

Pressure - closure strain response

Figure 3.13-4 shows a typical pressure - closure strain response. The gasket is loaded to point B, then unloaded to point C. When the closure strain becomes negative, the response is governed by $E_{tensile}$. Then the gasket is loaded to point D. Up to this time, the gasket behaves as a nonlinear elastic material. Then the gasket is loaded to point E and unloaded to point F. The unloading to point F is done on an unloading curve based at point E; this unloading curve is interpolated from the input unloading curves. The gasket is loaded to point G and unloaded to point H. The gasket follows an unloading curve based at point G and interpolated from the input unloading data. During the loading to point I, the gasket follows the unloading curve to point G, then follows the loading curve to point I. The gasket is then unloaded to points J and K. When the pressure becomes negative, the response is again governed by $E_{tensile}$. During reloading to point L, the gasket follows the same unloading curve as used for points J and K. The gasket is then loaded to point M, at which time the gasket ruptures. The pressure drops to zero and remains zero thereafter (point N).

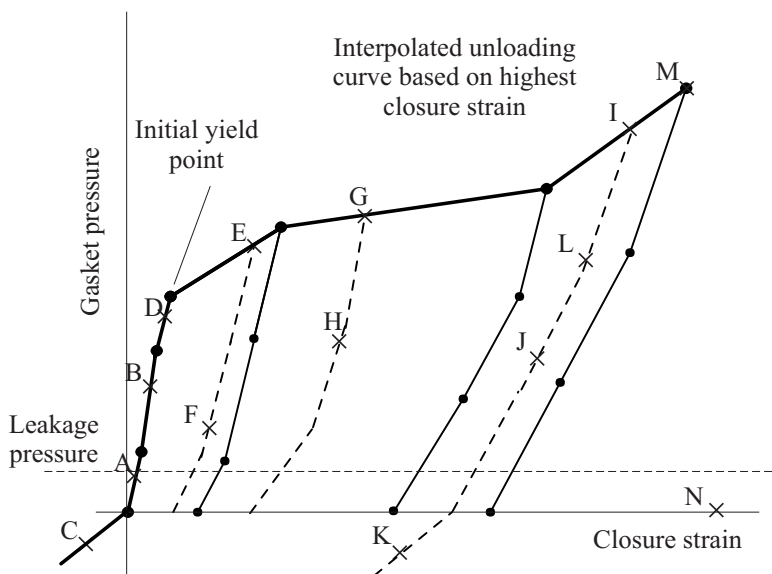
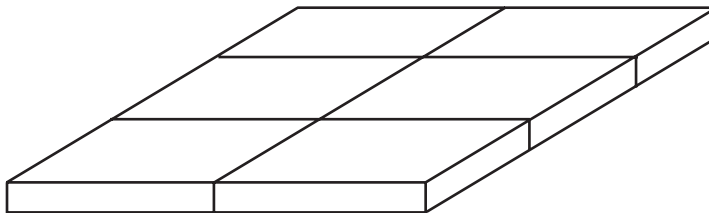


Figure 3.13-4: Typical pressure-closure strain history

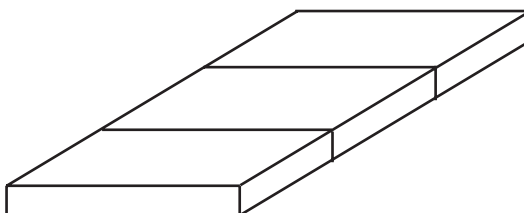
- Each gasket can have one of the following five states:
 - Open: The gasket pressure is less than the leakage pressure.
 - Closed: The gasket pressure is higher than the leakage pressure but has not yet caused plasticity.
 - Sealed: There has been plastic gasket deformation and the current pressure is above the gasket leakage pressure.
 - Leaked: After plastic deformation, the gasket pressure has dropped below gasket leakage pressure.
 - Crushed: Gasket closure strain has exceeded the rupture value.

Modeling issues

- The gasket must be modeled as a single layer of 2-D or 3-D elements. Linear elements are recommended (4-node quad elements in 2-D, 6-node prism and 8-node brick elements in 3-D). Higher order 2-D quad elements and 3-D prism and brick elements can also be used. In this case, the AUI automatically generates constraints for the midside nodes in the gasket direction to keep a linear thickness displacement variation.
- If the user does not explicitly set the material axes directions in gasket elements, the AUI attempts to automatically determine the gasket material axes directions. This automatic determination is made based on the element layout. Fig 3.13-5 shows element layouts in which the AUI can and cannot automatically determine the gasket material directions. If the AUI cannot automatically determine the gasket material directions, and if the material axes for the gasket are not explicitly defined by the user, then the AUI cancels dat file generation.



- (a) Each element is surrounded by neighboring elements on adjacent sides.
Gasket direction can be automatically determined.



- (b) At least one element is surrounded by neighboring elements on opposite sides only.
Gasket direction cannot be automatically determined.

Figure 3.13-5: Automatic determination of gasket material directions

- The AUI eliminates the in-plane deformations of simple gasket elements using nodal skew systems. The procedure is as follows. Skew systems are constructed with the skew a direction aligned with the gasket material direction, and the skew b and c directions aligned with the in-plane directions. The skew systems are assigned to the nodes of the simple gasket elements. Then the AUI deletes the b and c degrees of freedom at these nodes.

Note that this procedure is applied to all nodes attached to simple gaskets, even nodes that are also attached to other elements.

- It is allowed to deactivate the AUI's computation of the material axes directions and skew systems. This is done using command

MISC-OPTIONS GASKET-AUTOMATIC=NO

(the default is GASKET-AUTOMATIC=YES). When GASKET-AUTOMATIC=NO is used, the user must explicitly specify the

material axes directions. And, if simple gaskets are used, the user must explicitly eliminate the degrees of freedom corresponding to in-plane deformations.

- The top and bottom surfaces of a gasket can be separate from those of the mating surfaces. In this case, they should be connected via contact. The gasket can also share a common surface with the intended mating surface. In this case, contact is not needed, however, the gasket cannot separate from its target. A gasket can also be attached to its mating surface via tied contact, mesh glueing, constraint equations, or rigid links.

Output variables: The following gasket output variables are available for both gasket types: GASKET_PRESSURE, GASKET_CLOSURE_STRAIN, GASKET_THERMAL_STRAIN, GASKET_STRESS-*, GASKET_STRAIN-*, GASKET_THERMAL_STRAIN-*, GASKET_YIELD_STRESS, GASKET_PLASTIC_CLOSURE_STRAIN, GASKET_STATUS, GASKET_DEFORMATION_MODE

See Section 13.1.1 for the definitions of these variables. For the 2D gasket type, the components represented by the * are AA, BC and CC, and for 3D gasket type, the components represented by the * are BB, CC, AB, AC, BC.

3.14 Shape Memory Alloy (SMA) material model

- The Shape Memory Alloy (SMA) material model is intended to model the superelastic effect (SE) and the shape memory effect (SME) of shape-memory alloys.
- These materials can undergo solid-to-solid phase transformations induced by stress or temperature. The high temperature phase is called austenite (A) with a body-centered cubic structure and the low-temperature phase is called martensite (M) with a monoclinic crystal structure in several variants.

Fig.3.14-1 shows a schematic SMA stress-temperature

diagram. The martensite phase in two generalized variants and the austenite phase are shown, as well as stress- and temperature-dependent transformation conditions. The martensite phase is favored at low temperatures or high stresses. Upon heating from low temperature the material begins transforming from martensite to austenite at temperature A_s . The transformation is 100% complete at temperature A_f . If the material is then cooled again, the austenite starts transforming back to martensite at temperature M_s . This transformation is 100% complete at temperature M_f . These four temperatures are also stress dependent with high stresses favoring the martensite phase. This stress dependence is assumed to be linear with slope C_M and C_A for the martensite and austenite temperatures, respectively. A typical variation of volume fraction of martensite in the SMA material with temperature is shown in Fig. 3.14-2.

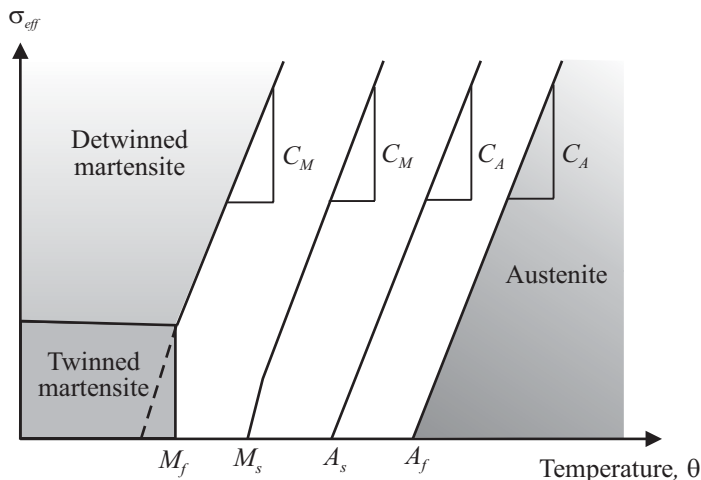


Figure 3.14-1: SMA stress-temperature phase diagram

- A typical uniaxial isothermal stress-strain curve is shown in Fig. 3.14-3.
- The superelastic effect is evident when the material is

deformed at temperature $\theta > A_f$ and is displayed in Fig.3.14-3(a). The stress cycle application induces transformations from A \rightarrow M and then from M \rightarrow A to exhibit the hysteresis loop. The shape memory effect is evident when the material is deformed at temperature $\theta < A_s$ and is displayed in Fig.3.14-3(b). A residual transformation strain remains after unloading; however heating the material to temperature above A_f leads to thermally induced M \rightarrow A transformation and the recovery of transformation strain.

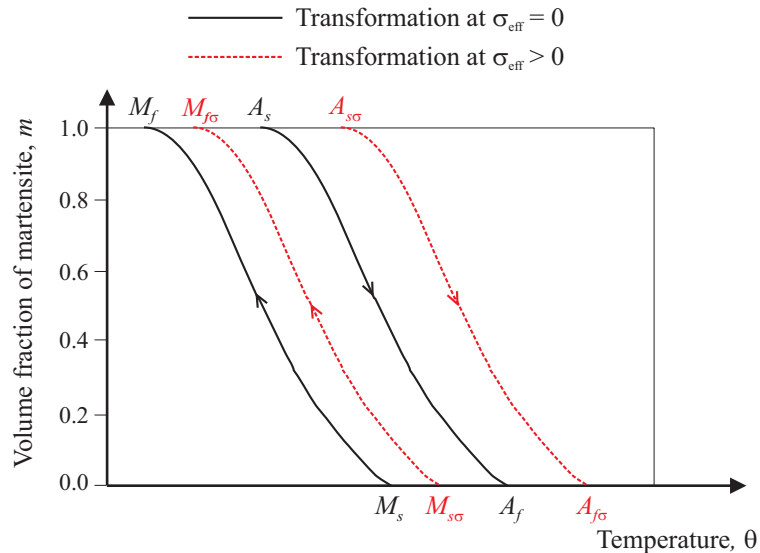


Figure 3.14-2 : Volume fraction of martensite vs. temperature

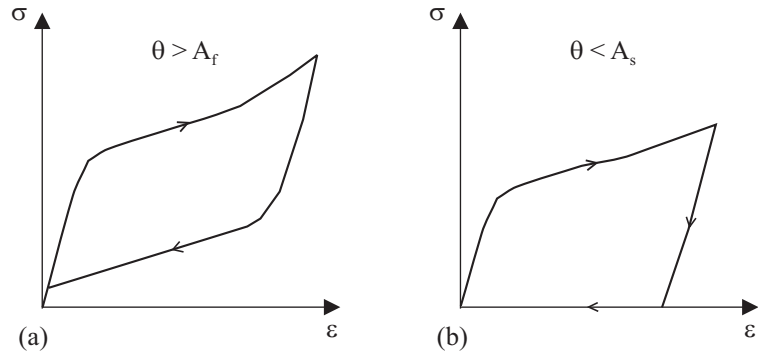


Figure 3.14-3: Schematic of stress-strain curves for shape-memory alloys:
(a) superelasticity; and (b) shape memory effect

- Both shape memory effects due to transformation from martensite to austenite and due to re-orientation of the martensite are captured by modeling the twinned and detwinned martensites as different phases.
- The SMA material model is based on the following equations:

- The total strain,

$$\mathcal{E} = \mathcal{E}^e + \mathcal{E}^t + \mathcal{E}^\theta$$

where

\mathcal{E}^e = elastic strain

\mathcal{E}^θ = thermal strain

\mathcal{E}^t = transformation strain; to be evaluated

- The one-dimensional macro-scale model,

$$\xi = \xi_s + \xi_l; \quad 0 \leq \xi \leq 1$$

$$\xi + \xi_A = 1$$

$$\varepsilon^t = \varepsilon_{\max}^t \xi_s$$

$$\sigma = ((1 - \xi)E_A + \xi E_M)(\varepsilon - \varepsilon^\theta - \varepsilon_{\max}^t \xi_s)$$

ξ_t = twinned martensite volume fraction

ξ_s = detwinned martensite volume fraction

ξ_A = austenite volume fraction

ε_{\max}^t = maximum recoverable residual strain; a material property usually obtained from a simple tension test when the material is fully detwinned martensite ($\xi_s = 1$)

► The flow rule of three-dimensional constitutive model,

$$\Delta \varepsilon_{ij}^t = \Delta \xi_s \varepsilon_{\max}^t n_{ij}^t$$

$$n_{ij}^t = \sqrt{\frac{3}{2}} \left(\frac{s_{ij}}{\bar{\sigma}} \right); \text{ for the martensitic transformation}$$

$$n_{ij}^t = \sqrt{\frac{3}{2}} \left(\frac{\varepsilon_{ij}^t}{\bar{\varepsilon}^t} \right); \text{ for the reverse transformation}$$

where

s_{ij} = deviatoric stresses

$$\bar{\sigma} = \sqrt{\frac{3}{2} s_{ij} s_{ij}} \text{ is the effective von Mises stress}$$

$$\bar{\varepsilon}^t = \sqrt{\frac{3}{2} \varepsilon_{ij}^t \varepsilon_{ij}^t} \text{ is the effective transformation strain}$$

This results in the following equation for deviatoric stress calculation:

$${}^{t+\Delta t}s_{ij} = \frac{{}^{t+\Delta t}E(\xi)}{1 + {}^{t+\Delta t}\nu(\xi)} \left({}^{t+\Delta t}\varepsilon_{ij}'' - \Delta\varepsilon_{ij}^t \right)$$

where

$${}^{t+\Delta t}\varepsilon_{ij}'' = {}^{t+\Delta t}\varepsilon_{ij}' - {}^t\varepsilon_{ij}^t$$

► Four phase transformation conditions,

1. Starting condition for the martensitic transformation

$$f_{M_s} = \sqrt{3J_2} - C_M(\theta - M_s)$$

2. Ending condition for the martensitic transformation

$$f_{M_f} = \sqrt{3J_2} - C_M(\theta - M_f)$$

3. Starting condition for the reverse transformation

$$f_{A_s} = \sqrt{3J_2} - C_A(\theta - A_s)$$

4. Ending condition for the reverse transformation

$$f_{A_f} = \sqrt{3J_2} - C_A(\theta - A_f)$$

► The phase transformation rate using linear kinetic rule (Auricchio and E. Sacco, 1999),

$$\Delta\xi = \frac{R_\xi}{f_f} \Delta f_f$$

$$\Delta f = \frac{3}{2} \frac{{}^{t+\Delta t}J_2 - {}^tJ_2}{{}^{t+\Delta t}\bar{\sigma}} - c({}^{t+\Delta t}\theta - {}^t\theta)$$

where, for the austenite to martensite transformation,

$$f_f = -f_{M_f}, \quad c = C_M \quad \text{and} \quad R_\xi = 1 - \xi$$

and for the reverse martensite to austenite transformation,

$$f_f = f_{A_f}, \quad c = C_A \quad \text{and} \quad R_\xi = \xi$$

► Evolution of single-variant detwinned martensite:
Martensite re-orientation is based on the following condition

$$f_R = \sqrt{3J_2} - C_R\theta - \sigma_R$$

where

$$\sigma_R = \text{material yield property at } \theta = 0$$

C_R = slope of yield function temperature variation

Austenite to martensite transformation leads to

$$\dot{\xi}_s = \dot{\xi}$$

Martensite to austenite transformation leads to proportional transformation of the twinned and detwinned phases:

$$\dot{\xi}_s = \frac{\xi_s}{\xi} \dot{\xi}$$

$$\dot{\xi}_t = \frac{\xi_t}{\xi} \dot{\xi}$$

- Computational steps for the stress-integration of the SMA model are as follows (Kojić and Bathe, 2005):

1. Calculate the trial deviatoric stresses, assuming no additional phase transformation or re-orientation,

$${}^{t+\Delta t}S_{ij}^{TR} = \frac{E({}^t\xi)}{1+\nu({}^t\xi)}({}^{t+\Delta t}\varepsilon_{ij}'')$$

2. Check for martensitic re-orientation,

$$f_R > 0 \text{ and } {}^t\xi_s < {}^t\xi \text{ and } {}^t\xi < 1$$

Check for austenite to martensite transformation,

$$f_{M_f} f_{M_s} < 0 \text{ and } {}^t\xi < 1 \text{ and } \Delta f > 0$$

Check for martensite to austenite transformation,

$$f_{A_f} f_{A_s} < 0 \text{ and } {}^t\xi > 0 \text{ and } \Delta f < 0$$

3. In case of martensitic re-orientation solve the following governing equation:

$$g_2(\Delta\xi_s) = \sqrt{\frac{3}{2} {}^{t+\Delta t}S_{ij} {}^{t+\Delta t}S_{ij}} - C_R {}^{t+\Delta t}\theta - \sigma_R = 0 \quad (3.14-1)$$

$${}^{t+\Delta t}S_{ij} = \frac{{}^{t+\Delta t}E(\Delta\xi)}{1+{}^{t+\Delta t}\nu(\Delta\xi)}({}^{t+\Delta t}\varepsilon_{ij}'' - \Delta\xi_s \cdot \varepsilon_{\max}^t {}^{t+\Delta t}n_{ij}^t)$$

The martensite reorientation calculation step is optional; it is activated when $\sigma_R > 0$ is input.

4. In case of austenite to martensite transformation, solve the following governing equation:

$$g(\Delta\xi) = \Delta\xi - \frac{{}^{t+\Delta t}R_\xi}{{}^{t+\Delta t}f_f} \left[\frac{3}{2} \frac{{}^{t+\Delta t}J_2(\Delta\xi) - {}^tJ_2}{{}^{t+\Delta t}\bar{\sigma}} - c({}^{t+\Delta t}\theta - {}^t\theta) \right] = 0 \quad (3.14-2)$$

where

$${}^{t+\Delta t}s_{ij} = \frac{{}^{t+\Delta t}E(\Delta\xi)}{1 + {}^{t+\Delta t}\nu(\Delta\xi)} ({}^{t+\Delta t}\varepsilon_{ij}'' - \Delta\xi_s(\Delta\xi) \cdot \varepsilon_{\max}^t {}^{t+\Delta t}n_{ij}^t)$$

$$R_\xi = 1 - \xi, \quad f_f = -f_{M_f} \quad \text{and} \quad c = C_M$$

5. In case of martensite to austenite transformation, solve the governing equation (3.14-2) with

$$R_\xi = \xi, \quad f_f = f_{A_f} \quad \text{and} \quad c = C_A$$

6. Update history-dependent variables for this time step/iteration step.

7. Calculate the consistent tangent constitutive matrix.

ref. M. Kojić and K.J. Bathe, *Inelastic Analysis of Solids and Structures*, Springer, 2005

ref. F. Auricchio and E. Sacco, “A Temperature-Dependent Beam for Shape-Memory Alloys: Constitutive Modelling, Finite-Element Implementation and Numerical Simulation”, *Computer Methods in Applied Mechanics and Engineering*, Vol. 174, pp. 171-190 (1999)

3.15 The Anand material model for soldering

- This material model is based on
 - ▶ Isotropic elasticity
 - ▶ Isotropic, isochoric viscoplastic flow
- The material model is described in the following references:

G. Weber and L. Anand, “Finite deformation constitutive equations and a time integration procedure for isotropic, hyperelastic-viscoplastic solids”, *Comp. Meth. Appl. Mech. Engng.*, 79 (1990) 173-202.

S. B. Brown, K. H. Kim and L. Anand, “An internal variable constitutive model for hot working of metals”, *Int. J. Plasticity*, 5 (1989) 95-130.

- The material model can be used with the **2-D plane strain**, **2-D axisymmetric** and **3-D solid** elements.
- The material model can be used with the **small displacement/small strain**, **large displacement/small strain** and **large displacement/large strain** formulations. When used with the small displacement/small strain formulation, a materially-nonlinear-only formulation is employed, when used with the large displacement/small strain formulation, the TL formulation is used, and when the large displacement/large strain formulation is used, the ULH formulation is used.
- If geometrically nonlinear effects are to be included, the large displacements/large strain kinematics are preferred to the large displacement/small strain kinematics, even when the strains are numerically small. The large displacement/small strain kinematics should be used only when the large displacement/large strain kinematics are not supported by the element. By default large strain kinematics are used when large displacement kinematics are selected.
- Plane strain, axisymmetric and 3-D elements that reference this material model should employ the mixed displacement-pressure (u/p) formulation. The u/p formulation is the default for these

elements.

- The material model does not possess a yield criterion, and no loading/unloading criteria. Instead, viscoplastic flow is assumed to take place at all levels of (deviatoric) stress, although at low levels, the rate of viscoplastic flow may be immeasurably small.
- The resistance to viscoplastic flow offered by the material is modeled using a single, scalar parameter s called the *isotropic deformation resistance*. This parameter has the dimension of stress.
- The material behavior is characterized by an evolution equation for the effective viscoplastic strain rate, and an evolution equation for the isotropic deformation resistance (Brown, et al.). These equations are

$$\dot{\epsilon}^{VP} = A_1 \exp\left(-\frac{Q}{R\theta}\right) \left[\sinh\left(\xi \frac{\sigma}{s}\right) \right]^{1/m} \quad (3.15-1)$$

$$\dot{s} = h_0 \left| \left(1 - \frac{s}{s^*}\right) \right|^{A_2} \operatorname{sgn}\left(1 - \frac{s}{s^*}\right) \dot{\epsilon}^{VP} \quad (3.15-2)$$

with

$$s^* = \hat{s} \left[\frac{\dot{\epsilon}^{VP}}{A_1} \exp\left(\frac{Q}{R\theta}\right) \right]^n \quad (3.15-3)$$

where A_1 is the pre-exponential factor, Q is the activation energy, R is the gas constant, θ is the absolute temperature, ξ is the multiplier of stress, m is the strain rate sensitivity of stress, h_0 is the hardening (or softening) constant, A_2 is the strain rate sensitivity of hardening (or softening), \hat{s} is the coefficient for the saturation value of the isotropic deformation resistance, n is the strain rate sensitivity of the deformation resistance, and σ is the equivalent tensile stress.

- The initial value of the isotropic deformation resistance s is s_0 .

- The material behavior is temperature dependent, but it can be made temperature independent by setting $Q = 0$.
- Thermal strains are calculated from the prescribed nodal point temperatures using a constant coefficient of thermal expansion.
- The Anand model is used in the modeling of solder materials.
- The uniaxial stress-strain response at room temperature is shown in Fig. 3.15-1 for a solder material with the following material constants:

$$E = 20 \cdot 10^3 \text{ MPa}$$

$$\nu = 0.31$$

$$A_1 = 2.3 \cdot 10^7 \text{ s}^{-1}$$

$$Q/R = 11262 \text{ K}$$

$$\xi = 11$$

$$m = 0.303$$

$$\hat{s} = 80.79 \text{ MPa}$$

$$n = 0.0212$$

$$h_0 = 4121.31 \text{ MPa}$$

$$A_2 = 1.38$$

$$s_0 = 42.32 \text{ MPa}$$

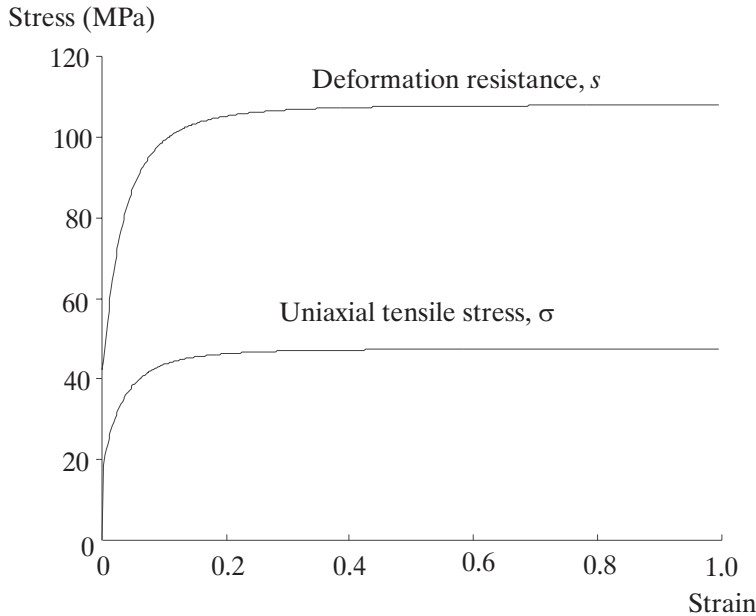


Fig. 3.15-1 The evolution of the isotropic deformation resistance in uniaxial tension

3.16 Piezoelectric material model

- The piezoelectric material model is available for 2-D and 3-D solid elements and can be applied to analyze the coupling behavior between the electric field and the corresponding mechanical deformation in structural analysis.
- The piezoelectric analysis discussed here is based on the direct coupling formulation, which is different than the iterative coupling formulation. Modeling with the iterative coupling formulation can set up a user-coded material (see Sections 3.17 and 12.4 for general information).
- In the direct coupling formulation, the electric potential is interpreted as a voltage degree of freedom (DOF), and each node of a solid element is assigned with a voltage DOF. In most cases, the number of the voltage DOFs in a solid element equals to the number of nodes in the element.

- For the 2D-solid element, the piezoelectric material model can be applied in the cases of plane strain, plane stress, axisymmetric, generalized plane strain and 3D plane stress. In the case of generalized plane strain, there is no voltage DOF assigned to the auxiliary node.
- The constitutive equation of the piezoelectric material model is considered to be linear. The TL formulation is available for piezoelectric analysis, which is thus applicable for large displacement/rotation problems.
- The piezoelectric material model can be used in static, dynamic-implicit, frequency, and mode superposition analysis. Currently, only the sparse solver is available for piezoelectric analysis. Due to the non-positive definite stiffness matrix of the piezoelectric material model, the setting of the control parameter "Continue Even When Non-Positive Definite Stiffness Matrix Encountered" in the AUI Control→Solution Process dialog box is ignored and the program always continues its solution.
- The piezoelectric material model cannot be used with:
 - ▶ explicit dynamic analysis
 - ▶ substructures
 - ▶ cyclic symmetry
 - ▶ mixed formulations

3.16.1 Constitutive equation

- The constitutive equation of the piezoelectric material model is

$$\begin{Bmatrix} \boldsymbol{\sigma} \\ \mathbf{D} \end{Bmatrix} = \begin{bmatrix} \mathbf{C} & \mathbf{e} \\ \mathbf{e}^T & -\boldsymbol{\epsilon} \end{bmatrix} \begin{Bmatrix} \boldsymbol{\gamma} \\ -\mathbf{E} \end{Bmatrix}$$

where

σ : elastic stress vector [force/area]

D : electric displacement field [electric charge/area]

C : elastic stiffness matrix [force/area]

e : piezoelectric coupling matrix [force/(volt·length)]

ϵ : dielectric matrix [electric charge/(volt·length)]

γ : elastic strain vector

E : electric field [volt/length]

For 2-D solid elements,

$$\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ D_a \\ D_b \end{Bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} & e_{1a} & e_{1b} \\ & C_{22} & C_{23} & C_{24} & e_{2a} & e_{2b} \\ & & C_{33} & C_{34} & e_{3a} & e_{3b} \\ & & & C_{44} & e_{4a} & e_{4b} \\ & \text{symm} & & & -\epsilon_{1a} & -\epsilon_{1b} \\ & & & & & -\epsilon_{2b} \end{bmatrix} \begin{Bmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \\ \gamma_4 \\ -E_a \\ -E_b \end{Bmatrix}$$

where the subscripts 1, 2, 3 and 4 respectively represent the local axes a , b , ab and c . There are totally 21 independent constants in the full constitutive matrix.

For 3-D solid elements,

$$\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \\ \sigma_6 \\ D_a \\ D_b \\ D_c \end{Bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} & e_{1a} & e_{1b} & e_{1c} \\ & C_{22} & C_{23} & C_{24} & C_{25} & C_{26} & e_{2a} & e_{2b} & e_{2c} \\ & & C_{33} & C_{34} & C_{35} & C_{36} & e_{3a} & e_{3b} & e_{3c} \\ & & & C_{44} & C_{45} & C_{46} & e_{4a} & e_{4b} & e_{4c} \\ & & & & C_{55} & C_{56} & e_{5a} & e_{5b} & e_{5c} \\ & & & & & C_{66} & e_{6a} & e_{6b} & e_{6c} \\ & & & & & & -\varepsilon_{aa} & -\varepsilon_{ab} & -\varepsilon_{ac} \\ & & & & & & & -\varepsilon_{bb} & -\varepsilon_{bc} \\ & & & & & & & & -\varepsilon_{cc} \end{bmatrix} \begin{Bmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \\ \gamma_4 \\ \gamma_5 \\ \gamma_6 \\ -E_a \\ -E_b \\ -E_c \end{Bmatrix}$$

symm

where the subscripts 1 - 6 respectively represent the local axes a , b , c , ab , ac and bc . There are totally 45 independent constants in the full constitutive matrix.

3.16.2 Orthotropic properties

- The piezoelectric material model is generally orthotropic. Therefore the elastic stiffness \mathbf{C} can also be defined in terms of the elastic modulus and Poisson's ratios (OPTION1 in command 'MATERIAL PIEZOELECTRIC') as follows:

For 2-D solid elements,

$$\mathbf{C}^{-1} = \begin{bmatrix} \frac{1}{E_a} & -\frac{\nu_{ab}}{E_b} & 0 & -\frac{\nu_{ac}}{E_c} \\ & \frac{1}{E_b} & 0 & -\frac{\nu_{bc}}{E_c} \\ & & \frac{1}{G_{ab}} & 0 \\ \text{symm} & & 0 & \frac{1}{E_c} \end{bmatrix}$$

For 3-D solid elements,

$$\mathbf{C}^{-1} = \begin{bmatrix} \frac{1}{E_a} & -\frac{\nu_{ab}}{E_b} & -\frac{\nu_{ac}}{E_c} & 0 & 0 & 0 \\ & \frac{1}{E_b} & -\frac{\nu_{bc}}{E_c} & 0 & 0 & 0 \\ & & \frac{1}{E_c} & 0 & 0 & 0 \\ & \text{symm} & & \frac{1}{G_{ab}} & 0 & 0 \\ & & & & \frac{1}{G_{ac}} & 0 \\ & & & & & \frac{1}{G_{bc}} \end{bmatrix}$$

- More information about the definition of the elastic-orthotropic material can be found in Section 3.2.2.

3.16.3 Strain-form coupling matrix

- The stress-form constitutive equation of the piezoelectric material model is often written in the strain-form

$$\boldsymbol{\sigma} = \mathbf{C}\boldsymbol{\gamma} - \mathbf{e}\mathbf{E} \Rightarrow \boldsymbol{\gamma} = \mathbf{C}^{-1}\boldsymbol{\sigma} + \mathbf{d}\mathbf{E}$$

where \mathbf{d} is called the strain-form piezoelectric coupling constants, and $\mathbf{e} = \mathbf{C}\mathbf{d}$.

- Thus, the piezoelectric coupling constants can be defined either in stress-form as \mathbf{e} , or in strain-form as \mathbf{d} (OPTION2 in command 'MATERIAL PIEZOELECTRIC'):

Matrix \mathbf{d} has a similar form as \mathbf{e} .

For 2-D solid elements,

$$\mathbf{d} = \begin{bmatrix} d_{1a} & d_{1b} \\ d_{2a} & d_{2b} \\ d_{3a} & d_{3b} \\ d_{4a} & d_{4b} \end{bmatrix}$$

and for 3-D solid elements,

$$\mathbf{d} = \begin{bmatrix} d_{1a} & d_{1b} & d_{1c} \\ d_{2a} & d_{2b} & d_{2c} \\ d_{3a} & d_{3b} & d_{3c} \\ d_{4a} & d_{4b} & d_{4c} \\ d_{5a} & d_{5b} & d_{5c} \\ d_{6a} & d_{6b} & d_{6c} \end{bmatrix}$$

3.16.4 Polarization direction of the piezoelectric material

- The coupling matrix \mathbf{e} (or \mathbf{d}) of a piezoelectric material model is generally not a full matrix due to the orthotropic property. Corresponding to different polarization directions, the nonzero entries are distributed in different places in the matrix \mathbf{e} (or \mathbf{d}). For example, in the 3-D case, if the polarization is along c , b and a direction respectively, the coupling matrix has the following form for the respective directions:

$$\mathbf{e} = \begin{bmatrix} 0 & 0 & e_{1c} \\ 0 & 0 & e_{2c} \\ 0 & 0 & e_{3c} \\ 0 & 0 & 0 \\ e_{5a} & 0 & 0 \\ 0 & e_{6b} & 0 \end{bmatrix}, \mathbf{e} = \begin{bmatrix} 0 & e_{1b} & 0 \\ 0 & e_{2b} & 0 \\ 0 & e_{3b} & 0 \\ e_{4a} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & e_{6c} \end{bmatrix} \quad \text{and}$$

$$\mathbf{e} = \begin{bmatrix} e_{1a} & 0 & 0 \\ e_{2a} & 0 & 0 \\ e_{3a} & 0 & 0 \\ 0 & e_{4b} & 0 \\ 0 & 0 & e_{5c} \\ 0 & 0 & 0 \end{bmatrix};$$

and in the 2-D case, if the polarization is along **a** and **b** direction respectively, the coupling matrix has the following form for the respective directions:

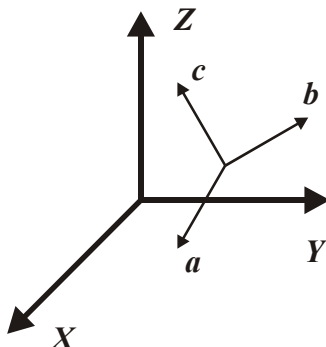
$$\mathbf{e} = \begin{bmatrix} 0 & e_{1b} \\ 0 & e_{2b} \\ e_{3a} & 0 \\ 0 & e_{4b} \end{bmatrix} \text{ and } \mathbf{e} = \begin{bmatrix} e_{1a} & 0 \\ e_{2a} & 0 \\ 0 & e_{3b} \\ e_{4a} & 0 \end{bmatrix}$$

- It is sometimes necessary to change the polarization from one direction to another. This is achieved by
 - Directly defining the coupling matrix **e** (or **d**) in its corresponding form as shown above. This is an easy operation only if the two polarizations are both along the coordinate axes.
 - For a more general case when the polarization is transformed between two arbitrary directions, the orthotropic axis system must be used, as discussed in next section.

3.16.5 Orthotropic axis system for the piezoelectric material model

For the 3-D solid element, if the local axes system is determined by three unit vectors **a**, **b** and **c** as shown in Figure 3.16-1, then the direction cosine matrix from local the global axes is defined as

$$\mathbf{A} = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} = \begin{bmatrix} a_x & a_y & a_z \\ b_x & b_y & b_z \\ c_x & c_y & c_z \end{bmatrix}$$

Figure 3.16-1: Material axes defined by unit vectors \mathbf{a} , \mathbf{b} and \mathbf{c}

Then the stress vector in the global system is calculated as

$$\hat{\boldsymbol{\sigma}} = \mathbf{A}_{\sigma} \boldsymbol{\sigma}$$

with

$$\mathbf{A}_{\sigma} = \begin{bmatrix} A_{11}^2 & A_{12}^2 & A_{13}^2 & 2A_{11}A_{12} & 2A_{11}A_{13} & 2A_{12}A_{13} \\ A_{21}^2 & A_{22}^2 & A_{23}^2 & 2A_{21}A_{22} & 2A_{21}A_{23} & 2A_{22}A_{23} \\ A_{31}^2 & A_{32}^2 & A_{33}^2 & 2A_{31}A_{32} & 2A_{31}A_{33} & 2A_{32}A_{33} \\ A_{11}A_{21} & A_{12}A_{22} & A_{13}A_{23} & A_{11}A_{22} + A_{12}A_{21} & A_{11}A_{23} + A_{13}A_{21} & A_{12}A_{23} + A_{13}A_{22} \\ A_{11}A_{31} & A_{12}A_{32} & A_{13}A_{33} & A_{11}A_{32} + A_{12}A_{31} & A_{11}A_{33} + A_{13}A_{31} & A_{12}A_{33} + A_{13}A_{32} \\ A_{21}A_{31} & A_{22}A_{32} & A_{23}A_{33} & A_{21}A_{32} + A_{22}A_{31} & A_{21}A_{33} + A_{23}A_{31} & A_{22}A_{33} + A_{23}A_{32} \end{bmatrix}$$

The strain vector in global system is calculated as

$$\hat{\boldsymbol{\gamma}} = \mathbf{A}_{\varepsilon} \boldsymbol{\gamma}$$

with

$$\mathbf{A}_{\varepsilon}^T = \mathbf{A}_{\sigma}^{-1}$$

The electric field and electric displacement field vectors in global

system are transformed as

$$\hat{\mathbf{E}} = \mathbf{A}\mathbf{E}$$

and

$$\hat{\mathbf{D}} = \mathbf{A}\mathbf{D}$$

The elastic stiffness matrix in global system is transformed from \mathbf{C}

$$\hat{\mathbf{C}} = \mathbf{A}_\sigma \mathbf{C} \mathbf{A}_\sigma^T$$

The piezoelectric coupling matrix in the global system is also transformed as

$$\hat{\mathbf{e}} = \mathbf{A}_\sigma \mathbf{e} \mathbf{A}^T$$

and

$$\hat{\mathbf{d}} = \mathbf{A}_\epsilon \mathbf{d} \mathbf{A}^T$$

For the 2-D solid element, the direction cosines can be simplified as

$$\mathbf{A} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

and the stress transformation matrix can be simplified as

$$\mathbf{A}_\sigma = \begin{bmatrix} c^2 & s^2 & -2cs & 0 \\ s^2 & c^2 & 2cs & 0 \\ cs & -cs & c^2 - s^2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where, $c = \cos \theta$ and $s = \sin \theta$.

Modeling hint:

In ADINA, the piezoelectric material properties are always defined in material local axes. You may need to rotate the material axes to the directions that you want them to be in the global axes by applying the orthotropic axis system. Figures 3.16-2(a) and (b) respectively show the local axis correspondence for 2D and 3D cases when the option 'Show Material Axes' is turned on in the AUI.

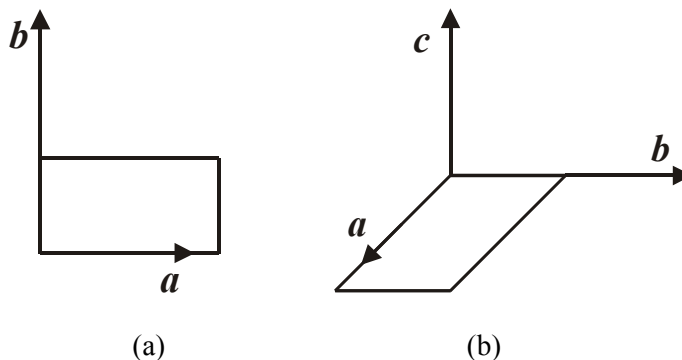


Figure 3.16-2: Material axes displayed in ADINA AUI.
(a) For 2D solid element (b) For 3D solid element

3.16.6 Electrical loading and boundary condition

- In piezoelectric analysis, the electric loading includes prescribed voltage, concentrated electric charge and distributed electric charge, which can be applied to 2-D and 3-D solid elements with the piezoelectric option. The voltage DOF is also one of the fixity options in the boundary condition definition.
- In 2-D analysis, the distributed charge can be applied on lines, edges, and element edge sets; in 3-D analysis, it can be applied on surfaces, faces, and element face sets. Both charge and voltage loads are scalar values without direction.
- It is recommended that at least one voltage DOF of each separate piezoelectric body be fixed or prescribed with a voltage value to suppress the numerical singularity.

3.17 Parallel-Network Framework

We now consider a material undergoing finite viscoelastic or elastoplastic motions modeled as N rheologically-parallel networks, each network potentially possessing both an elastic and inelastic response. For the purposes of this section, we will present the relevant continuum-mechanical quantities for each individual network.

- The various aspects of this framework are described in the following references:

- ref. M.E. Gurtin *et al.*, *The Mechanics and Thermodynamics of Continua*. Cambridge University Press, 2010.
- ref. E. Kröner, “Allgemeine Kontinuumsstheorie der Versetzungen und Eigenspannungen,” *Arch. Rat. Mech. Anal.* Vol. 4, pp 273–334, 1960.
- ref. E.H. Lee, “Elastic-plastic deformations at finite strains,” *ASME Journal of Applied Mechanics*, Vol. 35, pp 1–6, 1969.
- ref. F.J. Montáns and K.J. Bathe, “Computational Issues in Large Strain Elasto-Plasticity: An Algorithm for Mixed Hardening and Plastic Spin,” *Int. J. Numer. Meth. Engng.*, Vol. 63, pp 159–196, 2003.

3.17.1 Network-specific kinematics

Rheologically, the parallel network assumption requires that all networks undergo the same overall motions, giving rise to the kinematic constraint

$$\mathbf{X} = \mathbf{X}_i, \quad (3.17-1)$$

ref. KJB
Section 6.6.4

with $i = 1, \dots, N$. Furthermore, in this framework we adopt the approach of Kröner (1960) and Lee (1969) and multiplicatively decompose each network's deformation gradient tensor

$$\mathbf{X}_i = \mathbf{X}_i^e \mathbf{X}_i^v, \quad (3.17-2)$$

where \mathbf{X}_i^e and \mathbf{X}_i^v are tensors associated with the elastic and inelastic (viscous or plastic) parts, respectively. The inelastic part describes those motions that remain after the body is elastically unloaded; thus \mathbf{X}_i^v can be thought of as the motions from the reference configuration to an intermediate, unstressed ‘structural’ configuration. The elastic part \mathbf{X}_i^e defines motions from the intermediate configuration to the current configuration. Furthermore, $\det \mathbf{X}_i^v = 1$; that is, the inelastic motions are isochoric (volume-preserving). Thus, $J = J_i = \det \mathbf{X}_i^e$.

- This section generalizes for each of the N networks, thus we will suppress the subscript i for the remainder of the discussion.

The elastic right Cauchy-Green tensor and Green-Lagrange strain are

$$\mathbf{C}^e = \mathbf{X}^{eT} \mathbf{X}^e; \quad \mathbf{E}^e = \frac{1}{2}(\mathbf{C}^e - \mathbf{I}), \quad (3.17-3)$$

respectively. Inelastic strain and the total strain are thus equal when, at any instant, the body is unloaded (and unstressed).

The spatial velocity field is $\mathbf{v} = \dot{\mathbf{u}}$, and \mathbf{u} is the displacement field. The spatial velocity gradient $\mathbf{l} = \nabla \mathbf{v}$ can be written in terms of the elastic and inelastic distortions. We take the time derivative

$$\dot{\mathbf{X}} = \dot{\mathbf{X}}^e \mathbf{X}^v + \mathbf{X}^e \dot{\mathbf{X}}^v, \quad (3.17-4)$$

and since $\mathbf{X}^{-1} = \mathbf{X}^{v-1} \mathbf{X}^{e-1}$, we can write

$$\begin{aligned} \mathbf{l} &= (\dot{\mathbf{X}}^e \mathbf{X}^v + \mathbf{X}^e \dot{\mathbf{X}}^v) (\mathbf{X}^{v-1} \mathbf{X}^{e-1}), \\ &= \underbrace{\dot{\mathbf{X}}^e \mathbf{X}^{e-1}}_{\mathbf{l}^e} + \mathbf{X}^e \underbrace{\dot{\mathbf{X}}^v \mathbf{X}^{v-1}}_{\mathbf{l}^v} \mathbf{X}^{e-1}. \end{aligned} \quad (3.17-5)$$

Finally, and with no loss of generality,

$$\nabla \mathbf{v} = \mathbf{I} = \mathbf{I}^e + \mathbf{X}^e \mathbf{L}^v \mathbf{X}^{e-1}, \quad (3.17-6)$$

where \mathbf{I}^e is the spatial velocity gradient due to elastic motions and

$$\mathbf{L}^v = \mathbf{D}^v + \mathbf{W}^v, \quad (3.17-7)$$

is the structural velocity gradient for inelastic motions, \mathbf{D}^v is the inelastic deformation rate (the symmetric part of \mathbf{L}^v), and \mathbf{W}^v is the skew-symmetric inelastic spin tensor. Note that \mathbf{L}^v is deviatoric (its trace is zero), since $\det \mathbf{X}^v = 1$.

ref. KJB A key assumption within the framework for isotropic (visco)
Section 6.6.4 plasticity is that of *plastic irrotationality* – that the inelastic spin tensor \mathbf{W}^v vanishes. The irrotationality assumption allows us to write

$$\mathbf{L}^v = \mathbf{D}^v, \quad (3.17-8)$$

and recalling that $\mathbf{L}^v = \dot{\mathbf{X}}^v \mathbf{X}^{v-1}$,

$$\dot{\mathbf{X}}^v = \mathbf{D}^v \mathbf{X}^v. \quad (3.17-9)$$

Finally, the so-called flow direction is

$$\mathbf{N}^v = \frac{\mathbf{D}^v}{|\mathbf{D}^v|}, \quad (3.17-10)$$

and the inelastic deformation rate is related to the network-specific flow rule $\dot{\gamma}$ via

$$\mathbf{D}^v = \dot{\gamma} \mathbf{N}^v, \quad (3.17-11)$$

or, equivalently $|\mathbf{D}^v| = \dot{\gamma}$, where $\dot{\gamma}$ constitutively describes the network's time-dependent inelastic behavior.

3.17.2 Power expenditure, virtual velocities, and force balance

- This subsection draws heavily from the following reference:

ref. M.E. Gurtin *et al.*, *The Mechanics and Thermodynamics of Continua*. Cambridge University Press, 2010, sections 22, 84, and 91–101.

We can postulate a *power balance* of the form

$$\int_{\partial B} \mathbf{t}(\mathbf{n}) \cdot \mathbf{v} \, da + \int_B \mathbf{b} \cdot \mathbf{v} \, dv = \int_B \left(\mathbf{T}^e : \mathbf{I}^e + J^{-1} \mathbf{T}^v : \mathbf{L}^v \right) dv, \quad (3.17-12)$$

for a body B . The left and right sides of equation (3.17-12) represent external and internal power expenditures, respectively.

- $\mathbf{t}(\mathbf{n})$ is a surface traction, and \mathbf{n} is an outward unit normal to the boundary ∂B of body B .
- \mathbf{b} is a body force, possibly due to inertial effects.
- \mathbf{T}^e is the work conjugate of \mathbf{I}^e .
- \mathbf{T}^v is the work conjugate of \mathbf{L}^v .
- da and dv are differential area and volume elements, respectively, in the current configuration.

Note that all kinematic quantities are related by equation (3.17-6).

At this point, it is useful to introduce the concept of *virtual velocities*. Let us now consider \mathbf{v} , \mathbf{I}^e , and \mathbf{L}^v at an arbitrary, but fixed, instant. We will denote the virtual velocity $\tilde{\mathbf{v}}$ and virtual velocity gradients $\tilde{\mathbf{I}}^e$ and $\tilde{\mathbf{L}}^v$. These virtual quantities may be specified independently, though they must always satisfy the relationship (cf., equation 3.17-6)

$$\nabla \tilde{\mathbf{v}} = \tilde{\mathbf{l}}^e + \mathbf{X}^e \tilde{\mathbf{L}}^v \mathbf{X}^{e-1}. \quad (3.17-13)$$

Substituting these virtual quantities into equation (3.17-12) yields the so-called balance of virtual power

$$\int_{\partial B} \mathbf{t}(\mathbf{n}) \cdot \tilde{\mathbf{v}} \, da + \int_B \mathbf{b} \cdot \tilde{\mathbf{v}} \, dv = \int_B \left(\mathbf{T}^e : \tilde{\mathbf{l}}^e + J^{-1} \mathbf{T}^v : \tilde{\mathbf{L}}^v \right) dv. \quad (3.17-14)$$

Careful selection of admissible virtual velocities can yield powerful results. We will concentrate on two special cases:

- ▶ Zero virtual inelastic flow
- ▶ Zero virtual total velocity, but with internal relaxation
- Let us first specify that the virtual inelastic velocity gradient vanishes $\tilde{\mathbf{L}}^v = \mathbf{0}$. In this case, all virtual motions are due to the elastic response, equation (3.17-13) requires $\nabla \tilde{\mathbf{v}} = \tilde{\mathbf{l}}^e$, and the right hand side of equation (3.17-14) simplifies to $\int_B \left(\mathbf{T}^e : \nabla \tilde{\mathbf{v}} \right) dv$. The divergence theorem, substitution into (3.17-14) and some simplifications reveal that

$$\int_{\partial B} \left(\mathbf{t}(\mathbf{n}) - \mathbf{T}^e \mathbf{n} \right) \cdot \tilde{\mathbf{v}} \, da + \int_B \left(\nabla \cdot \mathbf{T}^e + \mathbf{b} \right) \cdot \tilde{\mathbf{v}} \, dv = 0, \quad (3.17-15)$$

and that $\mathbf{t}(\mathbf{n}) = \mathbf{T}^e \mathbf{n}$ and $\nabla \cdot \mathbf{T}^e + \mathbf{b} = \mathbf{0}$. Recalling Cauchy's postulate and the balance of linear momentum, it is clear that

$$\mathbf{T}^e \equiv \boldsymbol{\tau}, \quad (3.17-16)$$

which is the Cauchy stress.

We may now write the elastic stress-power as $\boldsymbol{\tau} : \mathbf{l}^e$. Since $\boldsymbol{\tau} = \boldsymbol{\tau}^T$, then $\mathbf{X}^{e-1} \boldsymbol{\tau} \mathbf{X}^{e-T}$ is also symmetric, by construction. Recalling that $\mathbf{l}^e = \dot{\mathbf{X}}^e \mathbf{X}^{e-1}$, one can define

$$\mathbf{S}^e \equiv J \mathbf{X}^{e^{-1}} \boldsymbol{\tau} \mathbf{X}^{e^{-T}} \quad (3.17-17)$$

as the so-called *elastic second Piola-Kirchhoff stress*.

- We will now select $\tilde{\mathbf{v}} = \mathbf{0}$ so that $\tilde{\mathbf{I}}^e = -\mathbf{X}^e \tilde{\mathbf{L}}^v \mathbf{X}^{e^{-1}}$. This corresponds to the special case of zero total velocities, but with virtual inelastic flow and internal stress relaxation. Equation (3.17-14) simplifies to

$$\int_B \left(\mathbf{T}^e : \tilde{\mathbf{I}}^e + J^{-1} \mathbf{T}^v : \tilde{\mathbf{L}}^v \right) dv = 0, \quad (3.17-18)$$

and we can write $J^{-1} \mathbf{T}^v : \tilde{\mathbf{L}}^v = \mathbf{T}^e : \left(\mathbf{X}^e \tilde{\mathbf{L}}^v \mathbf{X}^{e^{-1}} \right)$. Recalling that $\tilde{\mathbf{L}}^v$ is deviatoric and that $\mathbf{T}^e \equiv \boldsymbol{\tau}$ admits

$$\mathbf{T}^v = J \mathbf{X}^{e^T} \mathbf{dev}(\boldsymbol{\tau}) \mathbf{X}^{e^{-T}}, \quad (3.17-19)$$

where $\mathbf{dev}(\mathbf{A}) = \mathbf{A} - \frac{1}{3} \text{tr}(\mathbf{A}) \mathbf{I}$ is the deviator operator. Using the definition from equation (3.17-17), we can further simplify equation (3.17-19)

$$\begin{aligned} \mathbf{T}^v &= J \mathbf{dev} \left(\mathbf{X}^{e^T} \boldsymbol{\tau} \mathbf{X}^{e^{-T}} \right), \\ &= \mathbf{dev} \left(\left(\mathbf{X}^{e^T} \mathbf{X}^e \right) \mathbf{S}^e \right), \\ &= \mathbf{dev} \left(\mathbf{C}^e \mathbf{S}^e \right). \end{aligned} \quad (3.17-20)$$

The *Mandel stress* is

$$\mathbf{M} \equiv \mathbf{C}^e \mathbf{S}^e, \quad (3.17-21)$$

and $\mathbf{T}^v \equiv \mathbf{dev}(\mathbf{M})$. The deviatoric Mandel stress has thus been shown to be the work-conjugate of the inelastic velocity gradient, and therefore flow. Hence, the deviatoric Mandel stress is assumed

to drive inelastic flow and so (cf., equations 3.17-10 and 3.17-11)

$$\dot{\gamma} = \hat{\gamma}(\mathbf{dev}(\mathbf{M})); \quad \mathbf{N}^v = \frac{\mathbf{dev}(\mathbf{M})}{|\mathbf{dev}(\mathbf{M})|}. \quad (3.17-22)$$

It is important to note that the deviatoric Cauchy and Mandel stresses are identical for cases where the elastic and inelastic distortions are diagonal (as for uniaxial tension or compression and biaxial extension of isotropic materials) and when the motions are isochoric (strict incompressibility).

3.17.3 Energy dissipation

Viscoelastic/plastic motions are dissipative processes and thus involve hysteresis and associated energy dissipations. Within the context of a purely mechanical framework (i.e., neglecting temperature and entropy), the local free-energy inequality resulting from the first and second laws of thermodynamics can be written as

$$\left(\mathbf{S}^e - 2 \frac{\partial W(\mathbf{C}^e)}{\partial \mathbf{C}^e} \right) : \dot{\mathbf{E}}^e + \mathbf{dev}(\mathbf{M}) : \mathbf{D}^v \geq 0, \quad (3.17-23)$$

where W is the strain energy function constitutively describing the hyperelastic response, and $\dot{\mathbf{E}}^e = \mathbf{X}^{eT} \dot{\mathbf{X}}^e$ is the time rate of change of the elastic Green-Lagrange strain tensor. Assuming a purely

hyperelastic response $\mathbf{S}^e = 2 \frac{\partial W(\mathbf{C}^e)}{\partial \mathbf{C}^e}$, we arrive at the so-called reduced dissipation inequality

$$\mathcal{D}_{\text{int}} = \mathbf{dev}(\mathbf{M}) : \mathbf{D}^v \geq 0, \quad (3.17-24)$$

where \mathcal{D}_{int} is the internal energy dissipation.

3.17.4 Stress update procedure

Given the full deformation gradient tensor ${}^{t+\Delta t}_0 \mathbf{X}$ and the inelastic

deformation rate ${}^{t+\Delta t}\mathbf{D}^v$, integrating equation (3.17-9) over the time interval $[t, t + \Delta t]$ using a backward-Euler exponential mapping yields the updated inelastic distortion

$${}^{t+\Delta t}_0\mathbf{X}^v = \exp\left({}^{t+\Delta t}\mathbf{D}^v \Delta t\right){}_0^t\mathbf{X}^v, \quad (3.17-25)$$

from which we compute the elastic distortion

$${}^{t+\Delta t}_0\mathbf{X}^e = {}^{t+\Delta t}_0\mathbf{X}^v \left({}^{t+\Delta t}_0\mathbf{X}\right)^{-1}. \quad (3.17-26)$$

Recalling the definition of the Mandel stress and that

${}^{t+\Delta t}_0\mathbf{C}^e = \left({}^{t+\Delta t}_0\mathbf{X}^e\right)^T {}^{t+\Delta t}_0\mathbf{X}^e$, the updated Mandel stress and flow direction are

$$\begin{aligned} {}^{t+\Delta t}\mathbf{M} &= 2 {}^{t+\Delta t}_0\mathbf{C}^e \frac{\partial W\left({}^{t+\Delta t}_0\mathbf{C}^e\right)}{\partial {}^{t+\Delta t}_0\mathbf{C}^e}, \\ {}^{t+\Delta t}\mathbf{N}^v &= \frac{\mathbf{dev}\left({}^{t+\Delta t}\mathbf{M}\right)}{\left|\mathbf{dev}\left({}^{t+\Delta t}\mathbf{M}\right)\right|}, \end{aligned} \quad (3.17-27)$$

respectively. The correct Mandel stress satisfies

$$\hat{\gamma}\left(\mathbf{dev}\left({}^{t+\Delta t}\mathbf{M}\right)\right) \frac{\mathbf{dev}\left({}^{t+\Delta t}\mathbf{M}\right)}{\left|\mathbf{dev}\left({}^{t+\Delta t}\mathbf{M}\right)\right|} - {}^{t+\Delta t}\mathbf{D}^v = \mathbf{0}. \quad (3.17-28)$$

3.17.5 Three-network model

- This subsection describes ADINA's implementation, with some modifications, of the model described by:

ref. J.S. Bergström and J.E. Bischoff, "An Advanced Thermomechanical Constitutive Model for UHMWPE," *Int. J. Struct. Changes Sol.*, Vol. 2, pp 31–39, 2010.

The Three-network model is especially well-suited for capturing the highly nonlinear and time-, temperature-, and load history-dependent behaviors of synthetic polymers including Polytetrafluoroethylene (Teflon) and Polyethylene. In particular, it has the

ability to model viscoelastic effects, irreversible, load history-dependent changes to the material's effective stiffness, and entropic stiffening at high strains.

This model is implemented within ADINA's Parallel-Network framework and made available to the user via the command `MATERIAL THREE-NETWORK`.

- The Three-network model is available for the following elements types: 3D solid, 2D solid axisymmetric and plane strain, and 3D shell.

3.17.5.1 Rheological representation

The model's three networks A , B , and C are rheologically parallel and thus experience the same total motions

$$\mathbf{X} = \mathbf{X}_A = \mathbf{X}_B = \mathbf{X}_C. \quad (3.17-29)$$

That is, all three chains are constrained to move together. Network A captures viscoelastic effects, and network B captures both viscoelastic effects and irreversible material softening similar to permanent yielding. Finally, network C is purely hyperelastic and captures the effects of entropic stiffening at high strains.

- Note that although the effects of material softening depend on the loading history and are irreversible, this model is viscoelastic and not *plastic*. That is, inelastic flow may occur under any loading conditions, and there is no history-dependent “back-stress.” Thus, there is no permanent set.

3.17.5.2 Thermoelastic response

Networks A , B , and C all possess Eight-chain type hyperelastic responses. See section 3.8.1.6 for details. All three networks share the same locking stretch λ_L . The reduced network-specific elastic

chain stretch is $\bar{\lambda}_i = \sqrt{\frac{1}{3} \text{tr} \left(\det(\mathbf{C}_i^e)^{-1/3} \mathbf{C}_i^e \right)}$, where $i = A, B, C$.

When `THERMAL-EFFECTS=YES`, thermal effects are

incorporated directly through the shear modulus

$$\begin{aligned}\hat{\mu}_A(\mu_A, \theta) &= \mu_A \left(1 + \frac{\theta - \theta_0}{\hat{\theta}} \right), \\ \hat{\mu}_B(\mu_B, \theta) &= \mu_B \left(1 + \frac{\theta - \theta_0}{\hat{\theta}} \right), \\ \hat{\mu}_C(\mu_A, \theta) &= \mu_C \left(1 + \frac{\theta - \theta_0}{\hat{\theta}} \right),\end{aligned}\tag{3.17-30}$$

where θ is the temperature, θ_0 is a reference temperature, and $\hat{\theta}$ describes the temperature response. When temperature effects are not included $\hat{\mu}_A = \mu_A$, $\hat{\mu}_B = \mu_B$, and $\hat{\mu}_C = \mu_C$.

The user provides initial shear moduli (at the reference temperature) μ_A and μ_C directly. However, μ_B evolves according to the differential equation

$$\dot{\mu}_B = -\beta(\mu_B - \mu_{B_{\text{final}}})\dot{\gamma}_A,\tag{3.17-31}$$

with the initial condition $\mu_B = \mu_{B_{\text{init}}}$. The user supplies material parameters β , $\mu_{B_{\text{init}}}$, and $\mu_{B_{\text{final}}} \leq \mu_{B_{\text{init}}}$. This equation allows the model to capture irreversible changes to the material's effective stiffness. It also reveals that the elastic responses of networks A and B are not completely independent.

Given ${}^{t+\Delta t}\dot{\gamma}_A$, equation (3.17-31) is integrated over the time interval $[t, t + \Delta t]$ using a backward-Euler method to obtain the updated shear modulus

$${}^{t+\Delta t}\mu_B = \mu_{B_{\text{final}}} + \left({}^t\mu_B - \mu_{B_{\text{final}}} \right) \exp\left(-\beta {}^{t+\Delta t}\dot{\gamma}_A \Delta t\right),\tag{3.17-32}$$

and μ_B monotonically approaches $\mu_{B_{\text{final}}}$ as a function of network A 's inelastic flow rate history.

Network C is purely hyperelastic with reduced right and left Cauchy-Green tensors $\bar{\mathbf{C}} = J^{-2/3} \mathbf{X}^T \mathbf{X}$ and $\bar{\mathbf{b}} = J^{-2/3} \mathbf{X} \mathbf{X}^T$, respectively. Its deviatoric (volume-preserving) Cauchy stress contribution is

$$\begin{aligned} \boldsymbol{\tau}_{C_{\text{dev}}} = & \frac{1}{1+q} \left(\frac{\hat{\mu}_C(\mu_C, \theta)}{J \bar{\lambda}_C} \frac{\mathcal{L}^{-1}(\bar{\lambda}_C / \lambda_L)}{\mathcal{L}^{-1}(\lambda_L^{-1})} \text{dev}(\bar{\mathbf{b}}) \right) \\ & + \frac{q}{1+q} \left(\frac{\mu_C}{J} \left[J_1 \bar{\mathbf{b}} - \frac{2J_2}{3} \mathbf{I} - \bar{\mathbf{b}}^2 \right] \right), \end{aligned} \quad (3.17-33)$$

where $q \geq 0$ scales the effect of the first and second terms, \mathcal{L}^{-1} is the inverse of the Langevin function $\mathcal{L}(x) = \coth(x) - 1/x$, and

$$J_1 = \text{tr}(\bar{\mathbf{C}}); \quad J_2 = \frac{1}{2} \left[\text{tr}(\bar{\mathbf{C}})^2 - \text{tr}(\bar{\mathbf{C}}^2) \right], \quad (3.17-34)$$

are the first and second reduced invariants, respectively. The J_2 dependence in the second term of equation (3.17-33) phenomenologically captures the effects of entropic stiffening at high strains.

All three networks share the same bulk modulus κ , which enters the model through the volumetric strain energy contribution

$$W_V(J) = \kappa (J \ln J (J-1)). \quad (3.17-35)$$

The pressure is then $p = -\kappa \ln J$. The u/p formulation is always used with the Three-Network model.

3.17.5.3 Viscous response

The temperature-dependent viscous behavior of networks A and B is of the form

$$\dot{\gamma}_i = \dot{\gamma}_0 \left(\frac{M_i}{\hat{\tau}_i + a R(p_i)} \right)^{m_i} \left(\frac{\theta}{\theta_0} \right)^n, \quad (3.17-36)$$

for $i = A, B$. $M_i = \sqrt{\frac{1}{2} \mathbf{M}_{i_{\text{dev}}} : \mathbf{M}_{i_{\text{dev}}}}$ is the norm of the deviatoric Mandel stress $\mathbf{M}_{i_{\text{dev}}} = \mathbf{dev}(\mathbf{M}_i)$, p_i is the pressure, $\hat{\tau}_i$, a , and m_i are material constants governing the rate of viscous flow, $R(x) = \max(x, 0)$ is the ramp function, n is the temperature response exponential, and $\dot{\gamma}_0 = 1 \text{ s}^{-1}$ for dimensional consistency.

Note that although Bergström and Bischoff (2010) use the Cauchy stress to drive the viscous response, we use the Mandel stress for the reasons outlined in section 3.17.2.

Further, we take the plastic spin to vanish $\mathbf{W}_i^v = \mathbf{0}$, and thus the relation (cf., equation 3.17-6)

$$\mathbf{l} = \mathbf{l}_i^e + \mathbf{X}_i^e \mathbf{D}_i^v \mathbf{X}_i^{e-1}, \quad (3.17-37)$$

must hold at all times. For more information about these kinematic assumptions, see section 3.17.1.

Heat generation: A user-specified fraction of the energy dissipated by the viscous effects can be considered as heat generation. This heat generation can cause heating in a TMC (thermo-mechanical-coupling) analysis.

3.17.5.4 Constants and their interpretations

Three-Network model involves several material parameters, described below. Each parameter's corresponding designation in ADINA is given in parentheses and its dimensions in brackets, when applicable.

λ_L (LAMBDA_L): The locking stretch for all networks. λ_L must be greater than or equal to 1.0.

κ (KAPPA) $\left[\frac{\text{M}}{\text{L T}^2} \right]$: The bulk modulus for all networks.

a (PD): The pressure dependence. When $a = 0$, the viscous response is insensitive to the hydrostatic pressure. See also the parameter CALIBRATION, described in the section below

μ_A (MUA) $\left[\frac{\text{M}}{\text{L T}^2} \right]$: The initial shear modulus of network A at the reference temperature θ_0 .

$\hat{\tau}_A$ (FRA) $\left[\frac{\text{M}}{\text{L T}^2} \right]$: Resistance to viscous flow in network A .

As $\hat{\tau}_A \rightarrow \infty$, the viscous response in network A vanishes.

m_A (MA): Stress exponential for network A . When $m_A = 0$, the viscous response in network A is insensitive to the Mandel stress and hydrostatic stress.

$\mu_{B_{\text{init}}}$ (MUBI) $\left[\frac{\text{M}}{\text{L T}^2} \right]$: Network B 's initial shear modulus at the reference temperature and its virgin state.

$\mu_{B_{\text{final}}}$ (MUBF) $\left[\frac{\text{M}}{\text{L T}^2} \right]$: Network B 's final shear modulus. This must be less than or equal to $\mu_{B_{\text{init}}}$.

β (BETAB): This dimensionless constant controls how quickly the shear modulus for network B approaches $\mu_{B_{\text{final}}}$. Setting $\beta = 0$ recovers the special case of $\mu_B = \mu_{B_{\text{init}}}$ and completely decouples networks A and B .

$\hat{\tau}_B$ (FRB) $\left[\frac{M}{L T^2} \right]$: Resistance to viscous flow in network B .

m_B (MB): Stress exponential for network B .

μ_C (MUC) $\left[\frac{M}{L T^2} \right]$: Initial shear modulus for network C at the reference temperature.

q (QI2C): This positive, dimensionless constant governs the relative effects of the two terms of network C 's deviatoric stress. As $q \rightarrow 0$, the J_2 -dependence (the effects of entropic stiffening) vanishes.

$\hat{\theta}$ (TEMPHAT): Stiffness response to the change in temperature. As $|\hat{\theta}| \rightarrow \infty$, the shear modulus becomes insensitive to temperature. $\hat{\theta}$ must be nonzero.

θ_0 (TREF): Reference temperature. θ_0 must be greater than absolute zero, consistent with the temperature scale specified by TEMPUNIT.

n (TEMPN): Temperature exponential. When $n = 0$, the viscous response is insensitive to temperature.

- In addition to the 16 constants described above, the Three-Network model requires the following parameters.

TEMPUNIT: If the problem is to include temperature effects, the user may specify the temperature unit for TREF, TEMPHAT as Celsius, Fahrenheit, Kelvin, or Rankine.

CALIBRATION: By default, ADINA assumes that the material parameters are calibrated in accordance to the Three-network model defined in Bergström and Bischoff (2010). However, if the material parameters are calibrated using the MCalibration software developed by Veryst Engineering, then the user should specify CALIBRATION=MCAL. By doing so, the pressure dependence parameter a (PD) is adjusted to compensate for the

interpretation of the network-specific pressure in MCalibration. GENFAC: The heat generation factor (fraction of dissipation energy converted into heat generation). If GENFAC = 0, the dissipation is not calculated and there is no heat generation.

MAXITE: As discussed in section 3.17.4, Parallel Network framework requires an iterative material stress update to obtain the correct Mandel stresses and inelastic distortions. The parameter MAXITE specifies the maximum number of iterations that ADINA should use for this stress update convergence.

RTOL: The tolerance required for convergence of the stress update. It is not recommended to loosen this tolerance from the default of 1.0E-12.

3.17.5.5 Additional notes

- The network-specific pressure p_i is taken to be 1/3rd the total hydrostatic pressure. This is consistent with Bergström and Bischoff (2010). However, the MCalibration software interprets the network-specific pressure p_i to be *equal to* the total hydrostatic pressure. Thus, the user should specify CALIBRATION=MCAL when using parameters calibrated using the MCalibration software to make ADINA compensate for this discrepancy. Alternatively, the user can manually multiply the pressure dependence parameter a (PD) by 3.
- The total hydrostatic pressure cannot be accurately computed from the strains in the u/p formulation. Consequently, the stress update will not converge quadratically and the constitutive tensor will not be tangent.
- The Three-network model is highly sensitive to the stress exponentials m_A and m_B . Very large values for m_A and/or m_B may well slow or prevent material stress update convergence. In that case, the user may increase the number of material stress update iterations MAXITE and/or decrease the time step or use ATS. If at all possible, it is recommended that the user not loosen the stress update tolerance RTOL.

3.18 Cast iron material model

- This section describes the **thermo-cast iron** and **cast iron** material models, appropriate for modeling the elastoplastic response of grey cast iron. The discussion applies to both material models, unless otherwise noted.
- The cast iron material model is based on a composite yield criterion consisting of a Rankine surface in tension and the von Mises cylinder in compression. The cast iron material model uses an isotropic hardening rule.
- This material model can be used with the **2-D solid plane strain**, **2-D solid axisymmetric**, and **3-D solid** elements.
- This model can be used with the **small displacement/small strain**, **large displacement/small strain** and **large displacement/large strain** formulations. When used with the small displacement/small strain formulation, a materially nonlinear-only (MNO) formulation is employed; when used with the large displacement/small strain formulation, a TL formulation is employed; and when used with the large displacement/large strain formulation, a ULJ formulation is employed.
- If geometrically nonlinear effects are to be included, the large displacements/large strain kinematics are preferred to the large displacement/small strain kinematics, even when the strains are numerically small. The large displacement/small strain kinematics should be used only when the large displacement/large strain kinematics are not supported by the element. By default, large strain kinematics are used for 2-D solid and 3-D solid elements when large displacement kinematics are selected.
- Elements that reference this material model should also employ the mixed displacement-pressure (u/p) element formulation. The mixed formulation is the default for these elements.
- Under general yielding conditions, the cast iron material has a non-symmetric constitutive tensor. Therefore, to achieve optimal rates of convergence, the non-symmetric sparse solver is recommended when using the cast iron model.

3.18.1 Fundamental concepts

- The fundamental concepts of the cast iron material model are given in the following references:

ref. L. F. Coffin, "The Flow and Fracture of a Brittle Material," *J. Appl. Mech.*, 1950, Vol. 72, pp. 233–248.

ref. Shigley, *Mechanical Engineering Design*, 3rd edition, McGraw-Hill, 1977, Section 5.8.

- The cast iron model is motivated by the unique elastoplastic response of grey cast iron. Grey cast iron is known to be substantially more brittle than most other metals in tension, due to the presence of graphite flakes. These graphite flakes, embedded within the ferrite matrix, act as stress concentrators in tension. Conversely, these same graphite flakes serve to more evenly distribute stresses in compression. This microstructural feature results in markedly different responses in tension and compression, including different yield stresses, hardening curves. Also, although plastic motions in cast iron are volume-preserving (isochoric) under compression, they are not volume-preserving under tension.
- The elastic response in compression and tension is the same for the cast iron material model.

Interpolation of material constants as a function of

temperature: In the thermo-cast iron model, all material constants are functions of temperature. The material constants are entered for several temperatures as described in detail below.

During the solution process, the current temperature at an integration point is determined. Then the material constants are interpolated to the current temperature according to the following process:

Let ${}^{t+\Delta t}\theta$ be the current temperature, and θ_i and θ_{i+1} be two successive user-input temperatures that bracket ${}^{t+\Delta t}\theta$. A generic material constant, for example, the Young's modulus E , is specified at the two user-input temperatures: $E(\theta_i)$ and $E(\theta_{i+1})$, and

these specified values are interpolated to the current temperature using

$$E\left({}^{t+\Delta t}\theta\right)=E\left(\theta_i\right)+\left(E\left(\theta_{i+1}\right)-E\left(\theta_i\right)\right)\frac{{}^{t+\Delta t}\theta-\theta_i}{\theta_{i+1}-\theta_i}.$$

Yield condition: The cast iron material model uses a composite yield criterion consisting of a Rankine surface in tension and the von Mises cylinder in compression.

The yield condition in tension (Rankine) is:

$$F_t(\boldsymbol{\tau}, \sigma_{yt}) = \sigma_{pmax} - \sigma_{yt} \leq 0,$$

where σ_{yt} is the yield stress in tension and

$$\sigma_{pmax} = \max(\sigma_{p1}, \sigma_{p2}, \sigma_{p3}).$$

The yield condition in compression (von Mises) is:

$$F_c(\boldsymbol{\tau}, \sigma_{yc}) = \sigma_e - \sigma_{yc} \leq 0,$$

where σ_{yc} is the yield stress in compression.

These conditions can be visualized in the meridional plane, where

$$p = -\frac{\sigma_{p1} + \sigma_{p2} + \sigma_{p3}}{3}$$

is the pressure, and

$$q_1 = \sqrt{\frac{1}{2} \left[(\sigma_{p1} - \sigma_{p2})^2 + (\sigma_{p1} - \sigma_{p3})^2 + (\sigma_{p2} - \sigma_{p3})^2 \right]}$$

is the effective stress.

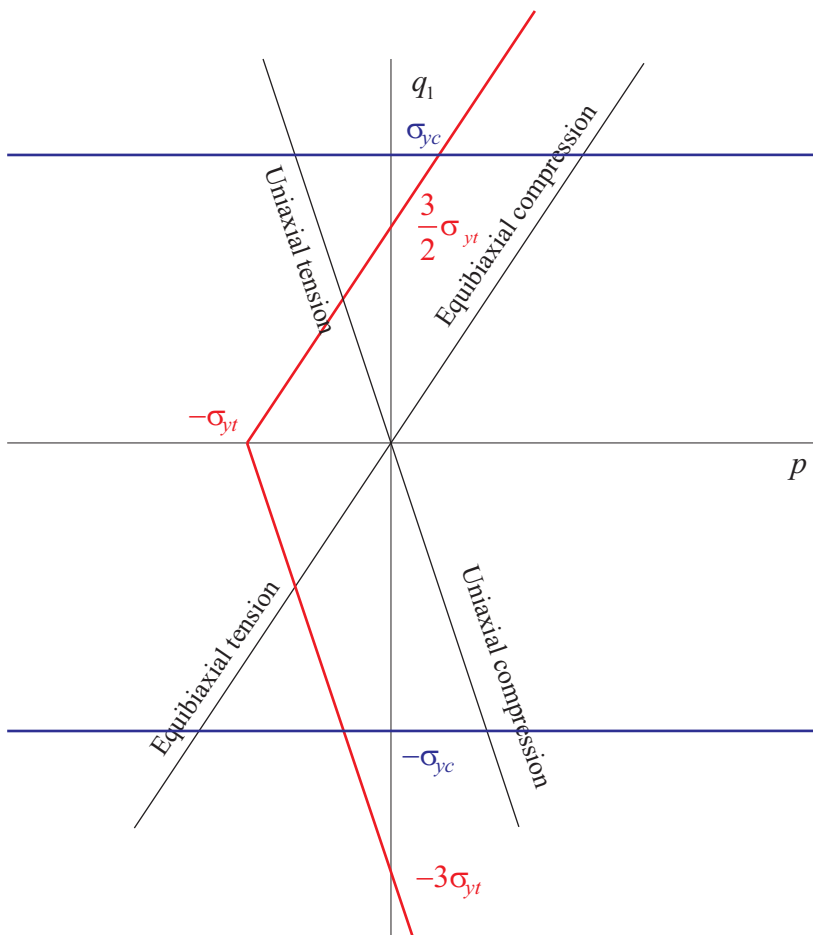


Figure 3.18.1: Visualization of yield conditions for cast iron in the meridional plane. The Rankine surface is shown in red, and the von Mises surface is shown in blue. The four loading conditions shown illustrate how the yield surfaces are intersected.

Note that it is possible for a material point to undergo plastic yielding in tension and later in compression (or vice-versa) during general loading conditions. It is also possible for a material point to undergo both tensile and compressive plastic yielding by following a loading path to where the Rankine and von Mises surfaces intersect.

Flow rule: The directions of the plastic strain increments are written in terms of a flow potential. This may be written

$$\dot{\mathbf{e}}^p = \dot{\lambda} \frac{\partial G}{\partial \boldsymbol{\tau}},$$

where G is the value of the flow potential and $\dot{\lambda}$ is a scalar multiplier. The flow potential is evaluated using a flow potential function:

$$g(p, \sigma_e, G) = 0.$$

The flow potential family of curves is parameterized by the value of p_0 , and therefore the flow potential value is $G = p_0$.

Again, we can visualize the flow potential in the meridional plane.

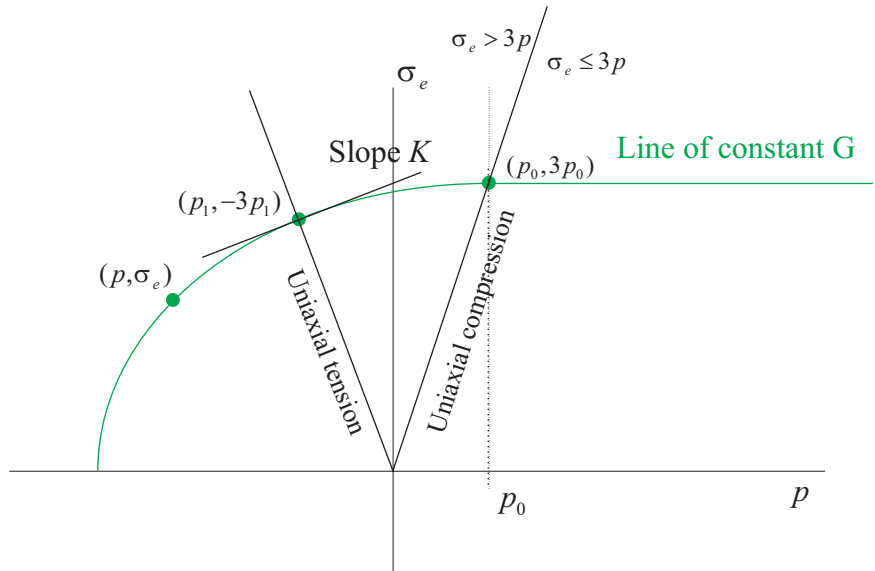


Figure 3.18.2: Visualization of the flow potential in the meridional plane. The slope K of the curve for uniaxial tensile loading is defined by the plastic Poisson's ratio ν_{pl} .

For $\sigma_e > 3p$, the flow potential is assumed to be an ellipse with center at $(p_0, 0)$, which passes through the points (p, σ_e) and $(p_0, 3p_0)$ and which has the specified slope

$$K = \frac{d\sigma_e}{dp} = \frac{3 - 6\nu_{pl}}{2(1 + \nu_{pl})},$$

where the uniaxial tension line intersects the ellipse. ν_{pl} is the user-defined plastic Poisson's ratio. The point where the uniaxial tension line intersects the ellipse is denoted $(p_1, -3p_1)$.

For $\sigma_e \leq 3p$, the flow potential is a horizontal line, corresponding to the von Mises surface. In this region, plastic motions are isochoric (volume preserving).

Isotropic hardening rule:

Multilinear hardening:

For each user-specified temperature, pairs of $(\bar{\epsilon}_c^P, \sigma_{y_c})$ and $(\bar{\epsilon}_t^P, \sigma_{y_t})$ are given, for the compressive and tensile response, respectively. It is allowed to have a different number of pairs for each specified temperature. The hardening curves for compressive and tensile behavior may also have different numbers of data pairs.

Then, for the current temperature $^{t+\Delta t}\theta$ and current accumulated effective compressive plastic strain $^{t+\Delta t}\bar{\epsilon}_c^P$ (for example), the current compressive yield stress $^{t+\Delta t}\sigma_{y_c}$ is determined by interpolation between the defined points on the multilinear curve.

Note that if the temperature suddenly changes, the yield stresses can also suddenly change.

Stress-strain integration: When plasticity is detected, the incremental plastic strains are solved for using an iterative solution

procedure.

Constitutive tensor: The constitutive tensor (stress-strain matrix) is constructed to be tangent. This gives good convergence in the equilibrium iterations. However, in general, the tangent constitutive tensor is nonsymmetric. Therefore, the non-symmetric sparse solver is recommended when using the cast iron model. The symmetric sparse solver may also be used; however, the constitutive tensor will then be symmetrized, reducing the rate of convergence (increase the number of equilibrium iterations).

Formulations: When used with the small displacement/small strain formulation, a materially-nonlinear-only (MNO) formulation is employed; when used with the large displacement/small strain formulation, a TL is employed; and when used with the large displacement/large strain formulation, a ULJ formulation is employed

Mixed displacement-pressure formulation: Elements that reference the cast iron material models should also employ the mixed displacement-pressure (u/p) element formulation.

Thermal strains: If a thermal load is applied to the structure, the thermal strains are taken into account. In the thermo-cast iron model, the coefficient of thermal expansion is temperature dependent. The procedure used is the same as for the thermo isotropic material model, see Section 3.5.

Modeling of failure: The cast iron model includes the modified Mohr failure criterion for brittle fracture. This criterion is described in the following reference:

ref. Shigley, *Mechanical Engineering Design*, 3rd edition, McGraw-Hill, 1977, Section 5.8.

In this theory, failure is governed by the maximum principal stress σ_{p1} and the minimum principal stress σ_{p3} . These stresses are compared against the user-supplied ultimate strengths in compression and tension S_{uc} and S_{ut} .

The safety factor n is the amount by which the stresses can be increased before the failure criterion is reached. Graphically, each safety factor n corresponds to a curve in the σ_{p1}, σ_{p3} plane, as shown.

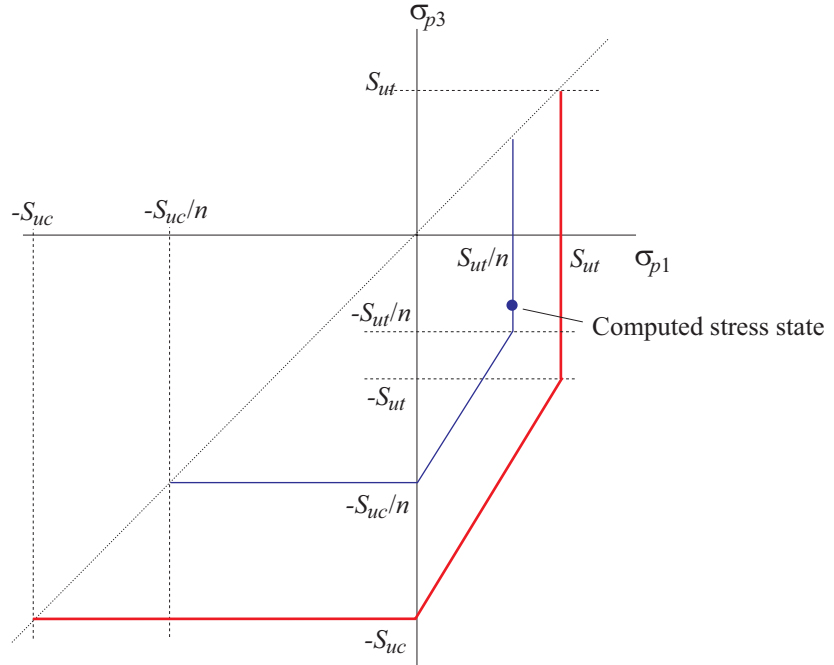


Figure 3.18.3: Failure envelope for the Modified Mohr Criterion for a specified safety factor n (shown in blue). The failure envelope for $n = 1$ is shown in red.

The computed safety factor n is calculated as followings:

- If $\sigma_{p3} \geq -\sigma_{p1}$, then $n = \frac{S_{ut}}{\sigma_{p1}}$
- If $\sigma_{p3} < -\sigma_{p1}$ and $\sigma_{p1} > 0$, then

$$n = \frac{S_{uc}}{\sigma_{p1} \left(\frac{S_{uc} - S_{ut}}{S_{ut}} \right) - \sigma_{p3}}$$

• If $\sigma_{p3} < -\sigma_{p1}$ and $\sigma_{p1} \leq 0$, then
$$n = \frac{-S_{uc}}{\sigma_{p3}}$$

When the computed safety factor is less than the user-input safety factor, failure is detected. When the stresses are zero, the computed safety factor is infinite, but the numerical output is 987654321.0. When plotting the computed safety factor, it may be useful to restrict the bandplot value range.

Modeling of rupture: Rupture conditions can also be included with this model. The user can specify one of two rupture options: MAXSTRAIN or FAILURE.

If the MAXSTRAIN option is selected, the material ruptures when the tensile accumulated effective plastic strain exceeds the maximum plastic strain from the user-input tensile stress-strain curve, or when the compressive accumulated effective plastic strain exceeds the maximum plastic strain from the user-input compressive stress-strain curve.

If the FAILURE option is selected, the material ruptures when failure is detected by the failure criterion. This option is allowed only when FAILURE=MODIFIED-MOHR.

When rupture is reached at a given element integration point, the corresponding element is removed from the model (see Section 11.5).

The option of user-supplied rupture is not available for the cast iron material.

Rate-dependent plasticity: The option of rate-dependent plasticity is not available for the cast iron model.

3.18.2 Specification of input

MATERIAL CAST-IRON: The basic command is

```
MATERIAL CAST-IRON NAME E NU DENSITY,  
                        ALPHA NUPL,  
                        SCURVE-C SCURVE-T,
```

FAILURE-CRITERION SAFETY-FACTOR,
SU-C SU-T RUPTURE-CRITERION

This command references data defined in command SCURVE, which defines stress-strain curves as piecewise linear through the data points. One curve defines the compressive response, and the other curve defines the tensile response.

The following conditions apply:

- 1) It is allowed to specify the same SCURVE definition for both tension and compression.
- 2) The stress-strain curves defined by the SCURVE commands are converted internally into stress-plastic strain curves.
- 3) After conversion of the stress-strain curves into stress-plastic strain curves, the successive plastic strains of the stress-plastic strain points must be increasing.
- 4) When RUPTURE-CRITERION=MAXSTRAIN, the material ruptures when the effective plastic strain exceeds the maximum plastic strain, not when the effective plastic strain exceeds the maximum user-input strain. The maximum plastic strain is obtained from the maximum user-input strain using the same conversion formula used to obtain plastic strains from user-input strains.
- 5) When using RUPTURE-CRITERION=MAXSTRAIN, in order to obtain a rupture condition in tension only, extend the compressive stress-strain curve so that the material cannot rupture in compression.

Input examples:

- 1) The simplest material that can be defined using this command is given by a command sequence such as

```
MATERIAL CAST-IRON NAME=1 E=17.4E4 NU=0.3,  
NUPL=0.04 SCURVE-C=1 SCURVE-T=2
```

This material references SCURVE=1 as the compressive hardening curve and SCURVE=2 as the tensile hardening curve. No failure or rupture criterion is included.

2) To include the modified Mohr failure criterion

```
MATERIAL CAST-IRON NAME=1 E=17.4E4 NU=0.3
NUPL=0.04,
SCURVE-C=1 SCURVE-T=2,
FAILURE-CRITERION=MODIFIED-MOHR,
SAFETY-FACTOR=3.0,
SUC=600.0 SUT=400.0
```

Note that the ultimate compressive and tensile stresses must also be included.

MATERIAL THERMO-CAST-IRON: The basic command is

```
MATERIAL THERMO-CAST-IRON NAME TREF DENSITY,
FAILURE-CRITERION SAFETY-FACTOR,
RUPTURE-CRITERION LIMITS-THETA
thetai Ei nui alphai nupli ,
scurve-compressioni,
scurve-tensioni su-ci
su-ti
```

The following conditions apply:

- 1) There must be at least two temperature rows in the MATERIAL THERMO-CAST-IRON command input. There is no limit on the number of temperature rows.
- 2) It is allowed to specify the same SCURVE definition more than once, for example, for the tension stress-strain curves at several temperatures, or for the tension and compression stress-strain curves at the same temperature.
- 3) The stress-strain curves defined by the SCURVE commands are converted internally into stress-plastic strain curves. All temperature interpolations are done using the stress-plastic strain curves.
- 4) After conversion of the stress-strain curves into stress-plastic strain curves, the successive plastic strains of the stress-plastic strain points must be increasing.

5) When RUPTURE-CRITERION=MAXSTRAIN, the material ruptures when the effective plastic strain exceeds the maximum plastic strain. For each user-input temperature, the maximum plastic strain is obtained from the maximum user-input strain using the same conversion formula used to obtain plastic strains from user-input strains. And the maximum plastic strain at an integration point is obtained from the maximum plastic strains at the user-input temperatures using temperature interpolation.

6) When using RUPTURE-CRITERION=MAXSTRAIN, in order to obtain a rupture condition in tension only, extend the compressive stress-strain curve so that the material cannot rupture in compression.

Input examples:

1) The following example includes the modified Mohr failure criterion and rupture. Cast iron material parameters are specified for 4 temperature (10, 50, 100, and 200 degrees), and the reference temperature is set to 100 degrees.

```
MATERIAL THERMO-CAST-IRON NAME=1 TREF=100.0,
          FAILURE-CRITERION=MODIFIED-MOHR,
          SAFETY-FACTOR=3.0,
          RUPTURE-CRITERION=FAILURE
10.0  17.3E4 0.29 1.0E-5 0.04 1 2 620.0 420.0
50.0  17.4E4 0.29 1.0E-5 0.04 3 4 610.0 410.0
100.0 17.5E4 0.29 1.0E-5 0.04 5 6 600.0 400.0
200.0 17.6E4 0.29 1.0E-5 0.04 7 8 500.0 300.0
```

Note that this example references 8 separate hardening curves – 2 at each temperature. Not all the material parameters in this example vary with temperature.

3.18.3 Output variables

The cast iron material outputs stresses, strains, plastic strains, accumulated effective compressive and tensile plastic strains

${}^{t+\Delta t}\bar{e}_c^P$, ${}^{t+\Delta t}\bar{e}_t^P$, compressive and tensile yield stresses ${}^{t+\Delta t}\sigma_{yc}$,

${}^{t+\Delta t}\sigma_{yt}$, and dissipated plastic work (defined as

$$w_P = \int_0^{t+\Delta t} \boldsymbol{\tau} : \mathbf{de}^P).$$

The cast iron material also outputs the current computed safety factor, the safety factor input, and the safety factor difference. A cast iron flag is also output showing if the material is yielding in compression and/or tension.

3.19 User-coded material model

- The user-coded material model is provided in ADINA to allow you to construct any type of material model for use with 1-D truss, 2-D or 3-D solid elements.
- This section first explains the general concepts involved in defining and using user-coded materials. The input and output parameters are then explained. This is followed by the details of the stress integration, then specific issues related to 1-D, 2-D and 3-D cases. Finally, the user-coded material models for which the source code is provided with the ADINA installation CD are described.

3.19.1 General considerations

- The constitutive relation must be arranged in the form of an algorithm and inserted into subroutine UCMAT1 for 1-D truss elements, subroutine UCMAT2 for 2-D solid elements or subroutine UCMAT3 for 3-D solid elements.
- The material model can be used with the following formulations:
 - ▶ Materially-nonlinear-only, for infinitesimally small displacements and strains (MNO)
 - ▶ total Lagrangian, in which the displacements and rotations can be very large, but the strains may have to be small or can be large depending on the material characteristics (TL)

- ▶ updated Lagrangian Hencky, in which case the strains can be large (ULH) (ULH formulation is not available for 1-D truss elements.)
- Either the displacement based elements or the mixed-interpolated displacement/pressure elements can be used with the user-coded material model. The mixed-interpolated elements are preferred for (nearly) incompressible materials. The mixed-interpolated element formulation is not applicable to 1-D truss elements.
- The user-supplied material subroutine UCMAT* is called for all integration points and is used to perform the following five types of operations during the various phases of the analysis. The type of operation to be performed is controlled by the integer variable KEY.

KEY = 0: Set size of working arrays (if requested). Set size of integer and real working arrays (LGTH2 and LGTH 1, respectively). Also select portion of these arrays to write the results to the (porthole) file (LGTH 4 and LGTH 3 for the integer and real arrays, respectively). This call with KEY = 0 is only performed if requested from the user-coded material command, by setting AUTOLEN = YES.

KEY = 1: Initialization call. Called once at the beginning of the analysis.

This call is for the initialization of the integration point variables, and is performed only once per integration point during the input phase. The arrays STRAIN, STRESS, ARRAY and IARRAY are automatically initialized by ADINA to zero. The user coding must set ARRAY and IARRAY to their proper initial values, if different from zero. Error checks on the material data and working array sizes should be performed here.

KEY = 2: Stress calculation call. Called for each iteration within a load (time) step. It is called for each subincrement if subincrementation is used.

This call is for the calculation of the element stresses at time $t + \Delta t$.

It is made during the solution of the governing equilibrium equations, and in the stress calculation phase after convergence is reached. In addition to the stress calculation, the integer and real history arrays IARRAY and ARRAY must also be updated when $KEY = 2$ (if they are used).

The stress integration can be performed in a single step using the total increment of strains and displacements or in subincrements of time. In this case, each subincrement spans a time of $\Delta t/n$ where n is the number of subincrements. This subincrementation may be useful for incremental rate type formulations such as viscoplasticity.

The actual stress integration adopted by the user within the increment or subincrement can be implicit or explicit. Details of the stress integration procedure are provided in Section 3.19.3.

If temperature-dependent material properties are present, the variables ALFA, ALFAA, CTD(*), CTDD(*) containing the properties at times t and $t + \Delta t$, are available for the user-supplied subroutine.

Note that if the ULH formulation is employed or if mixed-interpolated displacement/pressure elements are used, then the STRESS array will be internally modified by ADINA immediately after this stage to reflect the correct stresses at the element level.

KEY=3: Stress-strain matrix calculation calls. Called when the tangent stiffness matrix is to be calculated.

This call is for the calculation of the constitutive matrix **D** which is to be employed in the evaluation of the tangent stiffness matrix. The variables available for the constitutive matrix calculation are the values defined or calculated in the last subdivision of the stress calculation when $KEY=2$, as described above. Note that the algorithmic tangent constitutive matrix should be used if possible, to achieve quadratic convergence.

In the 1-D truss element case, **D** is the tangent modulus.

KEY=4: Stress printout call. Called for stress printout to the .out file.

This call is for the printing of the element response in the stress calculation phase. No stress calculations or updates should be performed when $KEY=4$.

Note that if the TL formulation is employed, ADINA automatically transforms, just for printout, the stresses used in the constitutive relations (see Table 3.19-1) to the Cauchy stresses in the global coordinate system before calling the user-supplied subroutine with KEY = 4.

In all cases of 1-D elements, the cross-sectional area of the element is assumed to remain unchanged, and the strain is equal to the longitudinal displacement divided by the original length.

If the ULH formulation is employed, total stretches PRST(3) are calculated and passed to the user-coded material subroutine.

In 2-D, PRST(1) stores the maximum in-plane stretch (maximum deviation from 1.0), PRST(2) stores the smaller in-plane stretch and PRST(3) is the stretch in the X-direction. Also, the angles of the first and the second principal stretches with respect to the Y-axis, stored in the array ANGLE(2), are passed to UCMAT2.

In 3-D, the PRST(3) vector stores the principal stresses and PHIST(3.3) stores the direction cosines.

Table 3.19-1: Stress and strain measures used in the user-coded material model. Note that the stresses are transformed internally by ADINA into true (Cauchy) stresses for printout (KEY=4)

Kinematic formulation	Strain measure	Stress measure
MNO (materially- nonlinear-only): small displacements, small strains	Infinitesimal strains	Cauchy stresses
TL (total Lagrangian): large displacements, small or large strains (depending upon the material model)	Green-Lagrange strains	2nd Piola-Kirchhoff stresses
ULH (updated Lagrangian Hencky): large displacements, large strains	Logarithmic Hencky (elastic) strain and plastic deformation gradient	Cauchy stresses

- The contents of the real and integer working array are written to the porthole file for post-processing at every integration point (the first LGTH3 entries for the real array, and the first LGTH4 entries for the integer array) . They can be accessed via the `USER_VARIABLE_I` and `INT_USER_VARIABLE_I` variables in ADINA-PLOT.

3.19.2 Input and output parameters

The user-supplied subroutines get numerous variables and arrays from ADINA to provide maximum flexibility to the user. The variables and arrays can be categorized as material properties, history dependent, algorithm control and output variables. Further documentation on these variables can be seen in the listings of the sample user-coded material subroutines provided in the ADINA installation. See, for example, the file `ovl30u_vp1.f`.

History dependent variables and arrays. Stored at integration points and characterize the history of the material behavior:

STRAIN(*): Strain components at time $t + \Delta t$. For the MNO and TL formulations, this vector stores the total strains (including elastic, inelastic, thermal). For the ULH formulation, STRAIN has the (right) elastic strains (total minus inelastic). The size of this vector is 4 for 2-D and 6 for 3-D. For the 1-D elements, STRAIN is the axial strain at time $t + \Delta t$.

EPS(*): Strain components at time t . For MNO and TL formulations, this vector stores the total strains. For the ULH formulation, it stores the (right) elastic strains at time t . For the 1-D elements, EPS is the axial strain at time t .

THSTR1(*): Thermal strain components at time t . For the 1-D elements, THSTR1 is the axial thermal strain at time t .

THSTR2(*): Thermal strain components at time $t + \Delta t$. For the 1-D elements, THSTR2 is the axial thermal strain at time $t + \Delta t$.

ARRAY(LGTH1): Working array of material history parameters (floating point variables).

LGTH1: Length of array ARRAY.

LGTH3: Number of terms in array ARRAY to save as results in porthole file.

IARRAY(LGTH2): Working array of material history parameters (integer variables).

LGTH2: Length of array IARRAY.

LGTH4: Number of terms in array IARRAY to save as results in porthole file.

PHIST(3,3): For the TL formulation, this array stores the components of the deformation gradient. For the ULH formulation, PHIST(3,3) stores the direction cosines corresponding to the principal (right) stretches.

PHIST1(3): For the ULH formulation, 3-D elements, and when KEY=4, this vector stores the direction cosines of the first principal (right) stretch (stretch which has the maximum deviation from 1.0).

RN(*,*): For the ULH formulation, this array stores the components of the rotation tensor (from the polar decomposition theorem).

PRST(3): For MNO, TL: Original coordinates of integration point.

For ULH: Principal right stretches.

Note that the PHIST, PHIST1, RN, and PRST arrays are not applicable to 1-D elements.

Note also that the ARRAY and IARRAY arrays are generated per integration point. Therefore, the size of these arrays should not be overestimated. Temporary local arrays should be used instead for temporary calculations if needed.

Material properties variables and arrays. These properties are provided via the user-coded material model in ADINA.

Three types of inputs are available for the user: a list of material constants, a list of solution control constants, and a table of all properties that are temperature dependent. The material and solution control constants serve the same purpose, and either of them can be used to provide the real-numbered material constants to the user-coded material. If, for example, the material used a Young's Modulus E which is constant it can be passed to the user-subroutine preferably in the material constant array. However, if E varies with temperature, it should be passed in the temperature dependent table. This information is passed to subroutine UCMAT* as follows:

CTI(NCTI): Array of material constants.

NCTI: Number of material constants (maximum 99).

SCP(NSCP): Array of solution control constants.

NSCP: Number of solution control constants (maximum 99).

CTD(NCTD): Temperature-dependent material properties constants for the given temperature at time t (τ for subincrementation). ADINA calculates CTD(NCTD) from the table of material parameters vs. temperature (if any) provided by the user-coded material model.

CTDD(NCTD): Temperature-dependent material properties constants for the given temperature at time $t + \Delta t$ (or $t + \Delta \tau$ for forward subincrementation). ADINA calculates CTDD(NCTD) from the table of material properties vs. temperature (if any) provided by the user-coded material model.

NCTD: Number of temperature-dependent material properties constants (maximum 98).

ALFA: Coefficient of thermal expansion for the given temperature at time t . ADINA calculates ALFA from the table of α vs. temperature (if any) provided by the user-coded

material model.

ALFAA: Coefficient of thermal expansion for the given temperature at time $t + \Delta t$ for forward stress integration or $t - \Delta t$ for backward stress integration. ADINA calculates ALFAA from the table of α vs. temperature (if any) provided by the user-coded material model.

Algorithm control variables. The algorithm control variables are passed to subroutine UCMAT* as follows:

TMP1: Temperature at time t .

TMP2: Temperature at time $t + \Delta t$ (forward stress integration) or $t - \Delta t$ (backward stress integration).

TIME: Time at $t + \Delta t$.

DT: Increment of time Δt .

INTEG: Flag for stress integration scheme. 0 = forward, 1 = backward.

ITE: Iteration number.

INTER: Number of subincrements for incremental stress integration.

Element information. These are the variables providing extra information about the element to subroutine UCMAT*.

NG: Element group number.

NEL: Element number.

IPT: Integration point number.

ETIMV: Birth time of the current element.

ETIMV2: Death time of the current element.

NODNUM(NNODE): Global node numbers of element NEL.

XYZ(*, NNODE): Initial nodal coordinates of element NEL.

NNODE: Number of nodes in element.

Output variables and arrays. These are variables that must be calculated by the user-supplied material at different phases of the analysis.

STRESS(*): The stress vector. Input to subroutine with stresses at time t and should output with stresses at time $t + \Delta t$. Calculated when KEY = 2. The size of this vector is 4 for 2-D and 6 for 3-D elements. For the 1-D elements, STRESS is the axial stress.

DP(*, *): Stress-strain matrix. Calculated when KEY = 3. The size of this array is 4×4 for 2-D and 6×6 for 3-D. This array is not applicable for 1-D elements.

DPSP(*): For the ULH formulation, this vector stores the increment of inelastic strains (plastic and/or creep and/or viscoplastic, etc.), calculated by the user-supplied code when KEY=2. This array is not applicable for 1-D elements.

3.19.3 Stress integration procedure

If the stress calculation at $t + \Delta t$ is solely based on the state at $t + \Delta t$,

$${}^{t+\Delta t}\boldsymbol{\sigma} = f\left({}^{t+\Delta t}\mathbf{e}, {}^{t+\Delta t}T, {}^{t+\Delta t}\alpha, \dots\right)$$

then it is the most straightforward to implement, and includes elastic and hyperelastic materials. However, for incremental or rate type functions, or such as

$$\dot{\boldsymbol{\sigma}} = f(\dot{\mathbf{e}})$$

the user must decide between single and multi-step calculation and also between implicit or explicit stress calculation. An example of an explicit stress calculation is

$${}^{t+\Delta t}\boldsymbol{\sigma} = {}^t\boldsymbol{\sigma} + {}^t\mathbf{C}^E \left(\Delta \mathbf{e} - \Delta \mathbf{e}^{th} - \Delta \mathbf{e}^{IN} \right) \quad (3.19-1)$$

An example of an implicit stress calculation is

$${}^{t+\Delta t}\boldsymbol{\sigma} = {}^{t+\Delta t}\mathbf{C} \left({}^{t+\Delta t}\mathbf{e} - {}^{t+\Delta t}\mathbf{e}^{th} - {}^{t+\Delta t}\mathbf{e}^{IN} \right) \quad (3.19-2)$$

where \mathbf{C}^E is the elastic constitutive matrix and $\Delta \mathbf{e}^{th}$ and $\Delta \mathbf{e}^{IN}$ are the increments of thermal and inelastic strains in the subincrement. For the 1-D elements, \mathbf{C}^E is the elastic modulus and $\Delta \mathbf{e}^{th}$ and $\Delta \mathbf{e}^{IN}$ are the increments of axial thermal strain and axial inelastic strains in the subincrement.

The calculation of Eq. (3.19-2) can be done iteratively or a predictor-corrector algorithm can be used (such as the elastic predictor followed by a radial vector corrector).

Backward stress integration

- The stress calculation is as follows:

First, the subroutine UCMAT* is called for the evaluation of the trial elastic stress $\boldsymbol{\sigma}_*^E$ corresponding to the total and thermal strains at time $t + \Delta t$, ${}^{t+\Delta t}\mathbf{e}$ and ${}^{t+\Delta t}\mathbf{e}^{th}$, respectively, and to the inelastic strains at time t , ${}^t\mathbf{e}^{IN}$. For example

$${}^{t+\Delta t}\boldsymbol{\sigma}_*^E = {}^{t+\Delta t}\mathbf{C}^E \left({}^{t+\Delta t}\mathbf{e} - {}^{t+\Delta t}\mathbf{e}^{th} - {}^t\mathbf{e}^{IN} \right) \quad (3.19-3)$$

where ${}^{t+\Delta t}\mathbf{C}^E$ is an elastic constitutive matrix corresponding to temperature ${}^{t+\Delta t}\theta$. Calculation of the trial elastic stress must be provided by your coding.

The stress correction $\Delta\sigma$ is then calculated. For example

$$\Delta\sigma = - {}^{t+\Delta t}\mathbf{C}^E \Delta\mathbf{e}^{IN}$$

where ${}^{t+\Delta t}\mathbf{C}^E$ is the elastic constitutive matrix corresponding to temperature ${}^{t+\Delta t}\theta$.

- Note that in the case of the updated Lagrangian Hencky formulation only the backward stress integration can be employed. Also, the increments of inelastic strains evaluated by your coding must be passed to ADINA for the calculation of the inelastic deformation gradient in ADINA (array DPSP).

- In the example of Eq. (3.19-3), arrays STRAIN and THSTR2 contain the total strains and thermal strains as described in the listing of subroutine UCMAT2 (see file ovl30u_vp1.f). Your coding must store into the working arrays ARRAY and/or IARRAY the history of deformation variables, e.g., the inelastic strains \mathbf{e}^{IN} , etc. and use them in the calculation of stresses.

In the calls to UCMAT* following calculation of the trial elastic stress, you must provide the coding for the stress increments due to the inelastic deformation.

Note that when using the ULH formulation, the array STRAIN contains the trial elastic strains for KTR = 0; the array is further updated in ADINA by subtracting increments of inelastic strains DPSP for each subdivision. Stresses and strains passed to subroutine UCMAT* correspond to deformation without rigid body rotations. The stresses STRESS which are true (Cauchy) stresses are passed to subroutine UCMAT* only for printout (KEY = 4). The transformation that takes into account the material rotation is performed in ADINA prior to calling subroutine UCMAT*.

An example of coding for the backward stress integration is given in the listing of the viscoplastic material example.

Note that when using the TL formulation, the stresses in array STRESS correspond to the second Piola-Kirchhoff stress tensor. The Cauchy stresses, obtained by the transformation of the second Piola-Kirchhoff stress tensor in ADINA, are passed to the subroutine UCMAT* only for printout (KEY=4).

If you require the stress invariants, your coding can call subroutine XJ1232 or XJ123 described in Section 3.19.4.

Note that if the large displacement/large strain options are used, stresses energetically conjugated with the corresponding strains (see Table 3.19-1) are employed in these calculations, and you cannot control this.

3.19.3.1 Subincrementation

Subincrementation can be used for forward stress integration. In this case, instead of going from time t to time $t + \Delta t$ in one step, subincrements are used. Subincrements can be forward in time from time t to $t + \Delta t$ in k steps as shown in Fig. 3.19-1 when $\Delta \tau = \Delta t/k$. The time forward subincrementation can be used with both implicit or explicit stress integration.

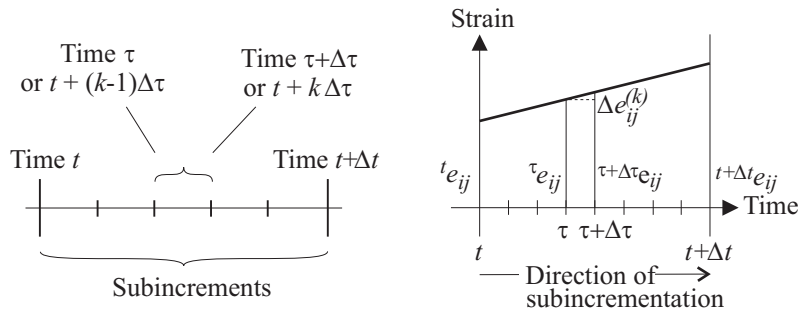


Figure 3.19-1: Schematic of forward time incrementation

	Values provided and passed back through the calling arguments of subroutine UCMAT*	
Variable or Array name	At entry, the arguments correspond to time:	At exit, the arguments correspond to time:
EPS(*)	t	t
STRAIN(*)	$t+\Delta t$	$t+\Delta t$
THSTR1(*)	τ	τ
THSTR2(*)	$\tau+\Delta\tau$	$\tau+\Delta\tau$
TMP1	τ	τ
TMP2	$\tau+\Delta\tau$	$\tau+\Delta\tau$
ALFA	τ	τ
ALFAA	$\tau+\Delta\tau$	$\tau+\Delta\tau$
CTD(*)	τ	τ
CTDD(*)	$\tau+\Delta\tau$	$\tau+\Delta\tau$
ARRAY(*)	τ	$\tau+\Delta\tau$
IARRAY(*)	τ	$\tau+\Delta\tau$
STRESS(4)	τ	$\tau+\Delta\tau$

No change in value

Figure 3.19-2: Variable definition for forward time incrementation

The variables passed to the user-coded material subroutines are defined as shown in Fig. 3.19-2.

If an explicit stress integration is employed, we get

$${}^{t+k\Delta t}\boldsymbol{\sigma} = {}^{t+(k-1)\Delta t}\boldsymbol{\sigma} + \Delta^k \boldsymbol{\sigma} \quad (3.19-4)$$

The stress increment $\Delta^k \boldsymbol{\sigma}$ in this example corresponds to strain and other quantities at time $t + (k-1)\Delta t$.

In the case of an implicit time integration, we can have

$$\Delta^k \boldsymbol{\sigma} = {}^{t+(k-1)\Delta t}\mathbf{C}^E \left(\Delta^k \mathbf{e} - \Delta^k \mathbf{e}^{th} - \Delta^k \mathbf{e}^{IN} \right) \quad (3.19-5)$$

where \mathbf{C}^E is the elastic constitutive matrix and $\Delta^k \mathbf{e}^{th}$ and $\Delta^k \mathbf{e}^{IN}$ are the increments of thermal and inelastic strains at time $t + k\Delta\tau$.

For the 1-D element case, equations (3.19-4) and (3.19-5) become scalar forms.

- The evaluation of the strain subincrements is performed by ADINA prior to calling subroutine UCMAT*. Your coding must update the stresses (in array STRESS) and other history dependent variables for each subincrement, and print the stresses once the final stresses for the load (time) step have been evaluated.
- The total strain increment is divided equally among the substeps.
- If temperature-dependent material properties are present, the variables ALFA, ALFAA, CTD(*), CTDD(*) containing the properties at times τ and $\tau + \Delta\tau$, are available for the user-supplied subroutine.

3.19.4 Coding guidelines

The ADINA user-coded materials use FORTRAN 90/95 conventions but maintain a few restrictions from FORTRAN 77 such as the 72-character line width.

The user is responsible for updating variables only at appropriate function calls (KEY = 2). It is advisable also to perform error checks during KEY = 1. These include checking the integrity of the material data input in the MATERIAL USER-SUPPLIED command, as well as the lengths of the working arrays assigned by the user. Errors can be reported to the .out file which is accessible via parameter IOUT which is passed to the user-coded material subroutine.

Several internal ADINA subroutines are available to facilitate the user implementation. These include:

SUBROUTINE XJ123 (STRESS, XI1, XI2, XI3, XJ1, XJ2, XJ3)

Variable descriptions:

Description:	Calculates the invariants of the 3-D stress tensor
Input:	Stress
Output:	XI1 — XI3. The 3 invariants of the stress tensor. XJ1 — XJ3. The 3 invariants of the deviatoric stress tensor.

They are calculated as follows:

$$\begin{aligned} \text{XI1} &= \tau_{ii}, & \text{XI2} &= 1/2 \tau_{ij} \tau_{ij}, & \text{XI3} &= 1/3 \tau_{ij} \tau_{jk} \tau_{ki} \\ \text{XJ1} &= 0, & \text{XJ2} &= 1/2 s_{ij} s_{ij}, & \text{XJ3} &= 1/3 s_{ij} s_{jk} s_{ki} \end{aligned}$$

where τ_{ij} is the stress tensor passed in STRESS and s_{ij} is the deviatoric stress tensor corresponding to τ_{ij} . The summation convention is used in the above expressions.

IMPLICIT DOUBLE_PRECISION (A-H, O-Z)
 DIMENSION STRESS (6)

SUBROUTINE XJ232 (STRESS, XI1, XI2, XI3, XJ1, XJ2, XJ3)

Variable descriptions:

Description: Calculates the invariants of the 2-D stress tensor
 Input: STRESS
 Output: XI1 — XI3. The 3 invariants of the stress tensor.
 XJ1 — XJ3. The 3 invariants of the deviatoric stress tensor.

They are calculated as in subroutine XJ123.

IMPLICIT DOUBLE_PRECISION (A-H, O-Z)
 DIMENSION STRESS (4)

SUBROUTINE BISECT (X, XMN, XPL, DX, XMIN, F, FMN, FPL, ACCEL, IMINUS, IACCEL, INDB)

Variable descriptions:

Description: Computes a new trial value of X and its

increment DX to find a solution for a nonlinear equation $F(X) = 0$ using the bisection method.

Input: F, ACCEL, IMINUS, IACCEL, INDB, described as follows:

F: current value of the function F

ACCEL: acceleration factor to increase DX

IMINUS: indicator for the slope of the function in the interval (XMN, XPL). Value of 1 for a positive slope, 2, for a negative slope.

IACCEL: indicator for application of acceleration factor during bisection inside the interval (XMN, XPL). Value of 0 when acceleration is not applied to the slope TT, and 1 when applied. Currently not used.

INDB: indicator, value of 1 when both XMN and XPL still not found, and value of 2 when XPL and XMN have been obtained.

Input & Output: X, FMN, FPL, XMN, XPL, described as follows:

X: current value of the argument X

FMN: negative value of F closest to 0

FPL: positive value of F closest to 0

XMN: X corresponding to FMN

XPL: X corresponding to FPL

Output: DX, described as follows:

DX: Increment of X (positive)

IMPLICIT DOUBLE_PRECISION (A-H, O-Z)

3.19.5 1-D truss elements

- The user-supplied material algorithm for 1-D elements must be placed in subroutine UCMAT1.
- A sample version of UCMAT1 may be found in file

ovl20u_vpl.f. The coding within this sample subroutine is fully explained in Section 3.19.8.1.

- The stress in this 1-D element is the axial stress.

3.19.6 2-D solid elements

- The user-supplied material algorithm for 2-D elements must be placed in subroutine UCMAT2.
- A sample version of UCMAT2 may be found in file ovl30u_vpl.f. The coding within this sample subroutine is fully explained in Section 3.19.8.1.
- The convention used for stresses and strains is shown in Fig. 3.19-3.

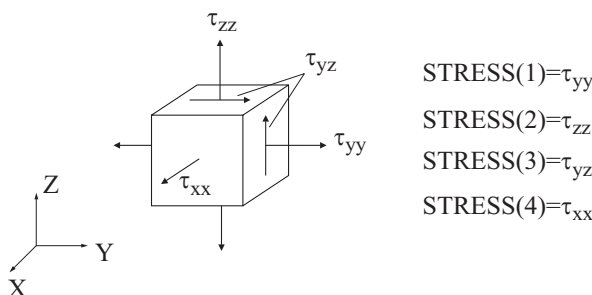


Figure 3.19-3: Stress convention for 2-D elements

3.19.7 3-D solid elements

- The user-supplied algorithm for the user-coded material model for 3-D solid elements must be placed in subroutine UCMAT3.
- A sample version of UCMAT3 may be found in file ovl40u_vpl.f. The coding within this sample version is fully discussed in the discussion of the viscoplastic model (see below).
- The convention used for stresses and strains is shown in Fig. 3.19-4.

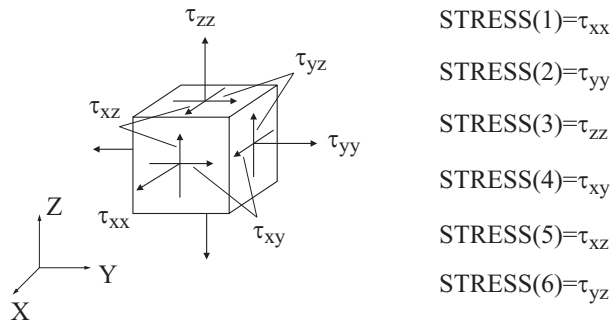


Figure 3.19-4: Stress/strain convention for 3-D elements

- When the ULH formulation is employed, the principal stretches and the direction cosines of the maximum principal stretch (with maximum deviation from 1.0) are passed to the subroutine UCMAT3 for the printout phase (KEY=4).
- If you require the stress invariants, your coding can call subroutine XJ123. This subroutine is described in Section 3.19.4. It returns the three invariants XI1, XI2 and XI3 of the stress tensor, and the three invariants XJ1, XJ2 and XJ3 of the deviatoric stress tensor.

3.19.8 User-coded material models supplied by ADINA R&D, Inc.

- User-coded material model subroutines are provided by ADINA (during installation) and can be modified by users for the following materials:
 - Viscoplastic
 - Elasto-plastic
 - Thermoplasticity and creep
 - Concrete with creep
 - Viscoelastic
 - Ramberg-Osgood with mixed hardening
 - Nonlinear elastic

- Mooney-Rivlin and Ogden

Details on some of these material models are provided in the sections below.

3.19.8.1 Viscoplastic material model

- A viscoplastic material model for axisymmetric or plane strain conditions is given as an example of the use of the UCMAT2 subroutine. A viscoplastic model for fully 3-D conditions is given as an example of the UCMAT3 subroutine, and a 1-D truss element example is also provided in the UCMAT1 subroutine of ovl20u.vol.f. The coding in subroutine UCMAT2 is based on the equations given hereafter.
- The UCMAT2 and UCMAT3 subroutine coding can be found in files ovl30u_vp1.f and ovl40u_vp1.f respectively. The UCMAT1 subroutine can be found in the file ovl20u.vol.f.
- The effective stress and accumulated effective plastic strain are stored as the first and second variables in the working array and provided as output in the porthole file. This is followed by the components of the viscoplastic strain tensor (4 in 2-D and 6 in 3-D). In the 1-D element case, the effective stress, accumulated effective plastic strain and viscoplastic strain are stored in the working array.
- The rate of viscoplastic strain is given by

$${}^t\dot{e}_{ij}^{VP} = \gamma < \phi > \frac{3}{2} \frac{{}^t s_{ij}}{{}^t \bar{\sigma}} \quad (3.19-6)$$

where

${}^t s_{ij}$ = component of deviatoric stress

${}^t \bar{\sigma} = \sqrt{\frac{3}{2} {}^t s_{ij} {}^t s_{ij}}$ = effective stress

γ = fluidity parameter

$$\begin{aligned} \langle \phi \rangle &= \phi & \text{if } \phi > 0 \\ &= 0 & \text{otherwise} \end{aligned}$$

In the above,

$$\phi = \frac{{}^t\bar{\sigma}}{\sigma_y} - 1 \quad (3.19-7)$$

where σ_y is the yield stress (above which the material is considered perfectly plastic).

- The index i has a range of 1 to 4 for 2-D and 1 to 6 for 3-D, and follows the conventions defined in Figures 3.19-3 and 3.19-4.
- The rest of this section assumes a 2-D material but all the expressions can be generalized to the 3-D case.
- The material is subjected to mechanical and thermal loading and the material constants are considered to be temperature independent.
- Stress integration is performed using forward or backward integration.

Forward time incrementation: In this case, the explicit Euler forward stress integration is used, where the increment of stresses in the subincrement (Eq. (3.19-2)) are

$$\Delta\sigma_i^{(k)} = C_{ij}^E \left(\Delta e_j^{(k)} - \Delta e_j^{th(k)} - \Delta e_j^{VP(k)} \right) \quad (3.19-8)$$

where

$$\Delta e_j^{(k)} = \frac{({}^{t+\Delta t}e_j - {}^te_j)}{\text{NSUBD}} \quad (3.19-9)$$

and NSUBD is the number of subincrements,

$$\Delta e_j^{th(k)} = \alpha \left({}^{\tau+\Delta\tau}\theta - {}^{\tau}\theta \right) \text{ for } j = 1, 2, 4$$

$$\Delta e_j^{VP(k)} = \frac{3}{2} (\Delta\tau) (\gamma) \left(\frac{{}^{\tau}\bar{\sigma}}{\sigma_y} - 1 \right) \frac{{}^{\tau}s_j}{{}^{\tau}\bar{\sigma}} \quad (3.19-10)$$

Here α is the mean coefficient of thermal expansion, τ and $\Delta\tau$ are defined in Fig. 3.19-1, e_j are the strains.

The same convention is used for increments of viscoplastic strains and deviatoric stresses. The coefficients C_{ij}^E represent the elastic constitutive matrix

$$\mathbf{C}^E = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1 & \frac{\nu}{(1-\nu)} & 0 & \frac{\nu}{(1-\nu)} \\ & 1 & 0 & \frac{\nu}{(1-\nu)} \\ & SYMM & \frac{(1-2\nu)}{2(1-\nu)} & 0 \\ & & & 1 \end{bmatrix} \quad (3.19-11)$$

where E and ν are the Young's modulus and Poisson's ratio.

In the 1-D case, equations (3.19-6), (3.19-8), (3.19-9), and (3.19-10) can be expressed as

$${}^t\dot{e}^{VP} = \gamma < \phi > \frac{3}{2} \frac{{}^t\sigma}{{}^t\bar{\sigma}} \quad (3.19-6a)$$

$$\Delta\sigma^{(k)} = \mathbf{C}^E \left(\Delta e^{(k)} - \Delta e^{th(k)} - \Delta e^{VP(k)} \right) \quad (3.19-8a)$$

$$\Delta e^{(k)} = \frac{\left({}^{t+\Delta t}e - {}^te \right)}{\text{NSUBD}} \quad (3.19-9a)$$

$$\Delta e^{th(k)} = \alpha \left({}^{\tau+\Delta\tau}\theta - {}^{\tau}\theta \right)$$

$$\Delta e^{VP(k)} = \frac{3}{2} (\Delta\tau) (\gamma) \left(\frac{{}^{\tau}\bar{\sigma}}{{}^{\tau}\sigma_y} - 1 \right) \frac{{}^{\tau}\sigma}{{}^{\tau}\bar{\sigma}} \quad (3.19-9a)$$

where C^E is the elastic modulus.

Backward time incrementation: In this case, the implicit Euler backward stress integration is used. The trial elastic state is determined according to Eq. (3.19-5),

$${}^{t+\Delta t}\sigma_{ij}^E = C_{ij}^E \left({}^{t+\Delta t}e_j - {}^{t+\Delta t}e_j^{th} - {}^te_j^{VP} \right) \quad (3.19-12)$$

The increments of stresses are obtained from

$$\Delta\sigma_i^{(k)} = -C_{ij}^E \Delta e_j^{VP(k)} \quad (3.19-13)$$

and the $\Delta e_j^{VP(k)}$ are given by Eq. (3.19-10).

If during subincrementation the effective stress $\bar{\sigma}^{(k)}$ is

$$\bar{\sigma}^{(k)} < \sigma_y \quad (3.19-14)$$

the subdivision loop is stopped by setting the parameter ISUBM to -1. Also the stresses are corrected as follows:

$${}^{t+\Delta t}\boldsymbol{\sigma} = -x \Delta\boldsymbol{\sigma}^{(k)} + \boldsymbol{\sigma}^{(k)} \quad (3.19-15)$$

where $x = \frac{\sigma_y - \bar{\sigma}^{(k)}}{\Delta\bar{\sigma}^{(k)}}$ and the ${}^{t+\Delta t}\boldsymbol{\sigma}$ are the final stresses at time $t+\Delta t$. Similarly, the increments of the viscoplastic strains are corrected using

$${}^{t+\Delta t}\mathbf{e}^{VP} = -x \Delta\mathbf{e}^{VP(k)} + \mathbf{e}^{VP(k)} \quad (3.19-16)$$

where ${}^{t+\Delta t}\mathbf{e}^{VP}$ are the final viscoplastic strains at time $t+\Delta t$.

In the 1-D element case, equations (3.19-12), (3.19-13), (3.19-15) and (3.19-16) are

$${}^{t+\Delta t}\sigma_*^E = C^E \left({}^{t+\Delta t}e - {}^{t+\Delta t}e^{th} - {}^te^{VP} \right) \quad (3.19-12a)$$

$$\Delta\sigma^{(k)} = -C^E \Delta e^{VP(k)} \quad (3.19-13a)$$

$${}^{t+\Delta t}\sigma = -x \Delta\sigma^{(k)} + \sigma^{(k)} \quad (3.19-15a)$$

$${}^{t+\Delta t}\mathbf{e}^{VP} = -x \Delta\mathbf{e}^{VP(k)} + \mathbf{e}^{VP(k)} \quad (3.19-16a)$$

Constitutive matrix: For the calculation of the constitutive matrix, the constitutive relation (3.19-8) can be written as

$$\Delta\sigma_m = c_m \left(\Delta e_m - \Delta e^{th} \right) \quad (3.19-17)$$

$$\Delta s_i = \frac{1}{a_E} \left(\Delta e'_i - \Delta e_i^{VP} \right) \quad (3.19-18)$$

where

$$\begin{aligned} \Delta e_m &= \frac{1}{3} (\Delta e_1 + \Delta e_2 + \Delta e_4) \\ \Delta e'_3 &= \frac{1}{2} \Delta e_3 \end{aligned} \quad (3.19-19)$$

are increments of the mean and deviatoric strains, and c_m and a_E are elastic constants

$$\begin{aligned} c_m &= \frac{E}{1-2\nu} \\ a_E &= \frac{1+\nu}{E} \end{aligned} \quad (3.19-20)$$

The derivatives of the mean stress and of the deviatoric stress follow from Eqs. (3.19-17) and (3.19-18):

$$\begin{aligned}\frac{\partial \sigma_m}{\partial e_j} &= \frac{1}{3} c_m \quad j = 1, 2, 4 \\ \frac{\partial s_i}{\partial e_j} &= \frac{1}{a_E} \left(\frac{\partial e'_i}{\partial e_j} - \frac{\partial (\Delta e_i^{VP})}{\partial e_j} \right)\end{aligned}\quad (3.19-21)$$

The derivatives $\frac{\partial e'_i}{\partial e_j}$ can be written in matrix form as

$$\frac{\partial e'_i}{\partial e_j} = \frac{1}{3} \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \quad i, j = 1, 2, 4 \quad (3.19-22a)$$

$$\frac{\partial e'_3}{\partial e_3} = \frac{1}{2} \quad (3.19-22b)$$

$$\frac{\partial e'_i}{\partial e_j} = 0 \quad \text{for } i = 3 \text{ or } j = 3 \text{ and } i \neq j \quad (3.19-22c)$$

In the case of viscoplastic flow in the time step, an approximate expression for $\frac{\partial (\Delta e_i^{VP})}{\partial e_j}$ can be obtained from Eq. (3.19-10) as

$$\frac{\partial (\Delta e_i^{VP})}{\partial e_j} = c^{VP} \frac{\partial s_i}{\partial e_j} \quad (3.19-23)$$

where

$$c^{VP} = \frac{3}{2} \frac{\Delta \tau}{t + \Delta t} \frac{\gamma}{\bar{\sigma}} \left(\frac{\bar{\sigma}^{t+\Delta t}}{\sigma_y} - 1 \right) \quad (3.19-24)$$

It follows from Eqs. (3.19-23) and (3.19-21) that the derivatives

$\frac{\partial s_i}{\partial e_j}$ are

$$\frac{\partial s_i}{\partial e_j} = c' \frac{\partial e'_i}{\partial e_j} \quad (3.19-25)$$

where

$$c' = \frac{1}{a_E + c^{VP}} \quad (3.19-26)$$

The constitutive matrix \mathbf{C}^{VP} can be obtained using the relation

$$\begin{aligned} \sigma_i &= s_i + \sigma_m \quad i = 1, 2, 4 \\ \sigma_3 &= s_3 \end{aligned} \quad (3.19-27)$$

from which it follows that

$$C_{ij}^{VP} = \frac{\partial s_i}{\partial e_j} + \frac{\partial \sigma_m}{\partial e_j} \quad i = 1, 2, 4 \quad (3.19-28a)$$

$$C_{3j}^{VP} = \frac{\partial s_3}{\partial e_j} \quad (3.19-28b)$$

By substituting Eqs. (3.19-21) and (3.19-25) into (3.19-28) and using Eq. (3.19-22), the constitutive matrix \mathbf{C}^{VP} is obtained as

$$\mathbf{C}^{VP} = \begin{bmatrix} \frac{1}{3}(c_m + 2c') & \frac{1}{3}(c_m - c') & 0 & \frac{1}{3}(c_m - c') \\ & \frac{1}{3}(c_m + 2c') & 0 & \frac{1}{3}(c_m - c') \\ \text{SYMM} & & \frac{1}{2}c' & 0 \\ & & & \frac{1}{3}(c_m + 2c') \end{bmatrix}$$

In the 1-D element case, the viscoplastic modulus is

$$C^{VP} = \frac{1}{a_E + c^{VP}}$$

3.19.8.2 Plasticity model

- An isothermal von Mises plasticity model with isotropic hardening is given as an example of the use of the UCMAT1, UCMAT2 and UCMAT3 subroutines.
- The implementation provided includes the MNO, TL and ULH formulations for 2-D and 3-D cases, and MNO and TL for the 1-D case.
- The UCMAT1, UCMAT2 and UCMAT3 subroutine coding can be found in files ovl20u_pl1.f, ovl30u_pl1.f and ovl40u_pl1.f respectively.
- The implementation is based on the equations given in Section 6.6.3 of Ref. KJB. The effective-stress-function algorithm is used for stress integration.

3.19.8.3 Thermo-elasticity and creep material model

- A model for thermoelasticity and creep is given as an example of the use of the UCMAT1, UCMAT2 and UCMAT3 subroutines. The model includes coding for the MNO, TL and ULH formulations for 2-D and 3-D cases, and MNO and TL for the 1-D

case.

- The UCMAT1, UCMAT2 and UCMAT3 subroutine coding can be found in files ovl20u_cn1.f, ovl30u_cn1.f and ovl40u_cn1.f respectively.
- The implementation is based on the equations given in Section 6.6.3 of ref. KJB.

The creep strain increments are defined using the Prandtl-Reuss equations:

$$d\varepsilon_{ij}^C = \gamma s_{ij}$$

where ε_{ij}^C are the creep strain components, s_{ij} are the deviatoric stress components and γ is the creep multiplier. Note that the effect of the hydrostatic pressure on the creep deformations is considered to be negligible.

The creep law is the eight-parameter creep law, i.e.:

$$\bar{\varepsilon}^C = f_1(\bar{\sigma}) f_2(\theta) f_3(t)$$

where $\bar{\varepsilon}^C$ is the effective creep strain, $\bar{\sigma}$ is the effective stress, θ is the temperature and t is the time, and

$$\begin{aligned} f_1(\bar{\sigma}) &= a_0 \bar{\sigma}^{a_1}; & f_2(\theta) &= \exp(-a_7 / (\theta + 273.16)) \\ f_3(t) &= t^{a_2} + a_3 t^{a_4} + a_5 t^{a_6} \end{aligned}$$

The same creep law also is used in the standard ADINA creep material model (see Section 3.6).

For multiaxial creep deformation, the calculation of the stresses is based on the use of an effective stress versus effective (creep) strain curve. The curve is assumed to be the same as the uniaxial stress-strain curve of the material.

In case of cyclic loading, the origin of the creep strain tensor depends on the stress tensor reversal and is determined using the ORNL rule (subroutines XCRCY1, XCRCY2 and XCRCY3).

The effective-stress-function algorithm is used for stress integration (subroutines XEFSF1, XEFSF2 and XEFSF3). Either

the time hardening procedure or the strain hardening procedure can be used in the calculation of the creep multiplier (subroutines XGAMA1, XGAMA2 and XGAMA3). A root-finding algorithm without acceleration (subroutine UBSECT) is used for computation of both the pseudo time (strain hardening) and the effective stress.

The tangent constitutive matrix ${}^{t+\Delta t}C_{ij}^{EC} = \frac{\partial {}^{t+\Delta t}\sigma_i}{\partial {}^{t+\Delta t}\varepsilon_j}$, where

${}^{t+\Delta t}\sigma_i$ are the stress components and ${}^{t+\Delta t}\varepsilon_j$ are the total strain components, is calculated using an approximation based on the effective-stress-function algorithm (subroutines XELMA2 and XELMA3). The corresponding 1-D tangent modulus

${}^{t+\Delta t}C^{EC} = \frac{\partial {}^{t+\Delta t}\sigma}{\partial {}^{t+\Delta t}\varepsilon}$ is calculated in subroutine XELMA1.

3.19.8.4 Concrete material model, including creep effects

- A model for concrete is given as an example of the use of the UCMAT2 and UCMAT3 subroutines. The model includes coding for the MNO, TL and ULH formulations for 2-D and 3-D cases, and MNO and TL for 1-D cases.
- The UCMAT1, UCMAT2 and UCMAT3 subroutine coding can be found in files ovl20u_cn2.f, ovl30u_cn2.f and ovl40u_cn2.f respectively.
- This user-coded material model has been developed for the analysis of concrete structures, in which creep deformations depend on the loading, on material ageing, on the temperature, and possibly on other material or load related parameters.
- The creep strain increments are defined by the Prandtl-Reuss equations

$$d\varepsilon_{ij}^C = \gamma s_{ij}$$

where ε_{ij}^C are the creep strain components, s_{ij} are the deviatoric stress components and γ is the creep multiplier. Consequently,

note that the effect of the hydrostatic pressure on the creep deformations is considered to be negligible.

- A generalized creep law of the following form is assumed:

$$\bar{\varepsilon} = f_1(\bar{\sigma}) f_2(\theta) f_3(t) f_4(x) + f_5(\bar{\sigma}) f_6(\theta) f_7(t) f_8(x)$$

where $\bar{\varepsilon}$ is the effective creep strain, $\bar{\sigma}$ is the effective stress, θ is the temperature, t is the time, and x is a user-defined variable.

Appropriate coding for the creep member functions f_1 to f_8 is provided (subroutines XFNC1, XFNC2 and XFNC3). In addition, coding for the time derivatives of f_3 and f_7 is also provided (subroutines XDFNC1, XDFNC2 and XDFNC3).

- Two different time variables are used (see Figure 3.19-5): t = time since concrete was cast; t_0 = time since ADINA analysis (and presumably loading) started. The quantity $(t - t_0)$ represents the age of the concrete at the start of the finite element analysis. The quantity $(t - t_0)$ is stored as the first entry in the SCP array. Therefore, $t = t_0 + \text{SCP}(1)$.

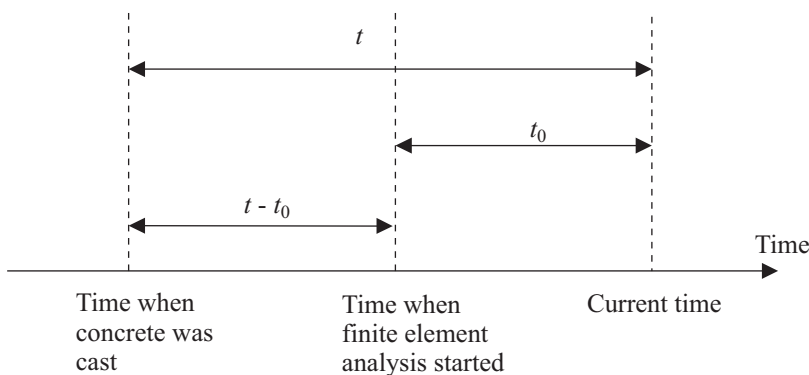


Figure 3.19-5: Time variables used for creep in concrete

- The second entry in the SCP array (SCP(2)) stores the flag for choosing between the strain hardening (value of 1) or the time hardening (value of 0) procedure in the effective-stress-function algorithm.
- The elastic modulus is assumed to be time dependent and given

by (see Figure 3.19-6):

$$E(t) = a_1 \exp \left[a_2 \left(1 - (a_3/t)^{a_4} \right) \right]$$

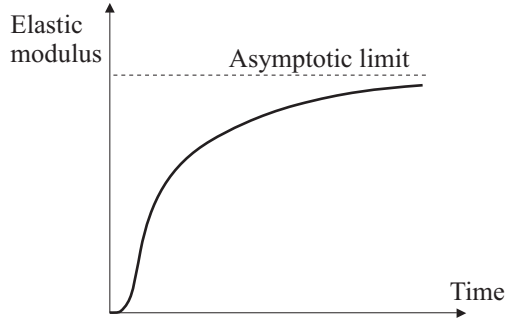


Figure 3.19-6: Time dependent elastic modulus for creep in concrete

- The creep law member functions are:

$$f_1(\bar{\sigma}) = \bar{\sigma}$$

$$f_2(\theta) = a_5 \phi^{a_6} + a_7 \phi^{a_8} \quad \text{with } \phi = \exp(a_9 \theta + a_{10})$$

$$f_3(t, t_0) = \frac{1}{E(t)} \left[\frac{a_{11} + t^{a_{12}}}{a_{13} + t^{a_{14}}} \right]^{a_{23}} \left[\frac{a_{15} + t_0^{a_{16}}}{a_{17} + t_0^{a_{18}}} \right]^{a_{24}} \left[\frac{a_{19} + (t - t_0)^{a_{20}}}{a_{21} + (t - t_0)^{a_{22}}} \right]^{a_{25}}$$

$$f_4(x) = 1$$

and

$$f_5(\bar{\sigma}) = \bar{\sigma}$$

$$f_6(\theta) = a_{26}(\theta + a_{27})^{a_{28}}$$

$$f_7(t) = \frac{1}{E(t)}$$

$$f_8(x) = 1$$

Note that the term of f_3 which depends only on $(t - t_0)$ could also be programmed as the member function $f_4(t - t_0)$.

- The creep constants are taken from array CTI as follows: $a_1 = \text{CTI}(1)$, $a_2 = \text{CTI}(2)$, One possible set of values is $a_1 = 2.0 \times 10^{10}$, $a_2 = 0.5$, $a_3 = 28$, $a_4 = 0.5$, $a_5 = 1.0$, $a_{12} = 0.6$, $a_{13} = 10$, $a_{14} = 0.6$, $a_{23} = 1.0$.
- In the case of multiaxial creep deformation, the calculation of the stresses is based on the use of an effective stress versus effective (creep) strain curve. The curve is assumed to be the same as the uniaxial stress-strain curve of the material.

In case of cyclic loading, the origin of the creep strain tensor depends on the stress tensor reversal and is determined using the ORNL rule (subroutines XCRCY1, XCRCY2 and XCRCY3).

The effective-stress-function algorithm is used for stress integration (subroutines XEFSF1, XEFSF2 and XEFSF3). The time hardening procedure or the strain hardening procedure can be used in the calculation of the creep multiplier, in case of variable stress and/or variable temperature conditions (subroutines XGAMA2 and XGAMA3). A root-finding algorithm without acceleration (subroutine UBSECT) is used for computation of both the pseudo time (strain hardening) and the effective stress.

The tangent constitutive matrix ${}^{t+\Delta t}C_{ij}^{EC} = \frac{\partial {}^{t+\Delta t}\sigma_i}{\partial {}^{t+\Delta t}\epsilon_j}$, where

${}^{t+\Delta t}\sigma_i$ are the stress components and ${}^{t+\Delta t}\epsilon_j$ are the total strain components, is calculated using an approximation based on the effective-stress-function algorithm (subroutines XELMA1, XELMA2 and XELMA3). The tangent modulus in the 1-D element

$$\text{case is } {}^{t+\Delta t}C^{EC} = \frac{\partial {}^{t+\Delta t}\sigma}{\partial {}^{t+\Delta t}\varepsilon}$$

3.19.8.5 Viscoelastic material model

- A model for linear viscoelasticity is given as an example of the UCMAT2 and UCMAT3 subroutines. The model includes coding for the MNO, TL and ULH formulations for 2-D and 3-D cases, and for the MNO and TL formulations for the 1-D case.
- The UCMAT1, UCMAT2 and UCMAT3 subroutine coding can be found in files ovl20u_vel.f ovl30u_vel.f and ovl40u_vel.f respectively.
- The implementation is fully described in Section 3.11.
- Subincrementation in the viscoelastic material model is only possible with forward time incrementation.
- The material constants must be specified in the CT1 array of floating point material constants.

Constant, 1: η_G

Constant, 2: G_∞

Constant, 3: G_1

Constant, 4: β_1

...

Constant, $\left[2(\eta_G + 1) + 1\right]$: η_K

Constant, $\left[2(\eta_G + 1) + 2\right]$: K_∞

Constant, $\left[2(\eta_G + 1) + 3\right]$: K_1

Constant, $\left[2(\eta_G + 1) + 4\right]$: γ_1

...

Constant, $\left[2(\eta_G + \eta_K + 2) + 1\right]$: T_0

Constant, $\left[2(\eta_G + \eta_K + 2) + 2\right]$: α_0

Constant, $\left[2(\eta_G + \eta_K + 2) + 3\right]: C_1$

Constant, $\left[2(\eta_G + \eta_K + 2) + 4\right]: C_2$

and the total number of constants is $2(\eta_G + \eta_K + 2) + 4$. In addition, the user must set the length of the floating point working array to 70 or larger in 2-D solid elements and 90 or larger for 3-D solid elements.

In the 1-D element case, only one series of modulus and decay constants is needed for the property description, i.e.,

Constant: η_E

Constant: E_∞

Constant: E_1

Constant: γ_1

...

Constant, $\left[2(\eta_E + 1) + 3\right]: T_0$

Constant, $\left[2(\eta_E + 1) + 4\right]: \alpha_0$

Constant, $\left[2(\eta_E + 1) + 5\right]: C_1$

Constant, $\left[2(\eta_E + 1) + 6\right]: C_2$

The size of the work array is $10 + 4\eta_E$ for the 1-D elements.

3.19.8.6 Ramberg-Osgood material model with mixed hardening

- The Ramberg-Osgood material model with mixed hardening is provided as an example of the UCMAT2 and UCMAT3 subroutines. The model includes coding for the axisymmetric, plane strain, plane stress and 3-D elements and for the MNO, TL and ULH formulations. For 1-D elements, only the MNO and TL formulations are applicable.
- The UCMAT2 and UCMAT3 subroutine coding can be found in files ovl30u_pl2.f and ovl40u_pl2.f respectively.

- The Ramberg-Osgood stress-strain description of the yield curve is in the form

$${}^t\sigma = {}^0\sigma_y + C_y \left(\mathbf{e}^p \right)^n$$

where C_y and n are material constants and ${}^0\sigma_y$ is the initial yield stress.

- The mixed hardening behavior is described by

$$d\mathbf{e}^p = d\mathbf{e}^{p_i} + d\mathbf{e}^{p_k} = M d\mathbf{e}^p + (1-M) d\mathbf{e}^p$$

where $d\mathbf{e}^{p_i}$ and $d\mathbf{e}^{p_k}$ are, respectively, the isotropic and kinematic contributions, and the yield function of the mixed hardening model is

$${}^t f_y = \frac{1}{2} \left({}^t\mathbf{s} - {}^t\boldsymbol{\alpha} \right) \cdot \left({}^t\mathbf{s} - {}^t\boldsymbol{\alpha} \right) - \frac{1}{3} {}^t\sigma_y^2 = 0$$

where the back stress ${}^t\boldsymbol{\alpha}$ is evolved by

$$d\boldsymbol{\alpha} = C_p (1-M) d\mathbf{e}^p$$

C_p is Prager's hardening parameter and is calculated as

$$C_p = \frac{2}{3} \frac{E_p - M E_p^M}{1 - M}$$

where E_p^M is the derivative of the yield curve at the plastic strain

$M\mathbf{e}^p$, E_p is the plastic modulus corresponding to \mathbf{e}^p , and M is the factor used in general mixed hardening ($0 < M < 1$) which can be a variable, expressed as

$$M = M_\infty + (M_0 - M_\infty) \exp(-\eta e^p)$$

- The effective-stress-function algorithm is used to calculate the plastic multiplier, see subroutines XEFSA and XEFBS in 2-D (subroutine UCMAT2) and XEFSF3 in 3-D (subroutine UCMAT3). The corresponding subroutine is XEFS1 for the 1-D case.

3.19.8.7 Nonlinear elastic material model

- A model for nonlinear elasticity is given as an example of the UCMAT1, UCMAT2 and UCMAT3 subroutines which can be found in files ovl20u_ne.f ovl30u_ne.f and ovl40u_ne.f respectively. For the theory and algorithm of this material model, please refer to Section 3.3.
- The material constants must be specified in the Constant Properties table of the User-Coded Material dialog box of the AUI as follows:

Constant 1: ν

Constant 2: ε_1

Constant 3: σ_1

Constant 4: ε_2

Constant 5: σ_2

...

Constant (2*n): ε_n

Constant (2*n+1): σ_n

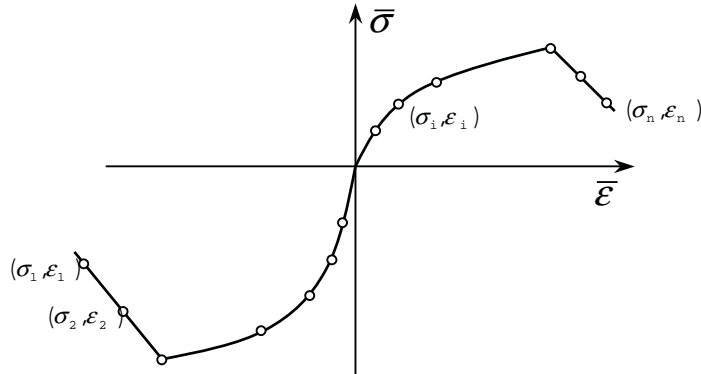


Figure 3.19-7: Input order of data point on stress-strain curve

- Note that discontinuities are not allowed in user-supplied stress-strain curve. The table look-up is performed using linear interpolation within the table and linear extrapolation outside the table using the two starting or ending points.
- In Solution Control table, the first row is a flag for using tangent or secant stress-strain matrix when the stress-strain curve enters into a softening region. The secant stress-strain matrix is employed for softening when this flag set to 1. Otherwise, the tangent stress-strain matrix will always be used.
- The backward stress integration scheme should be selected with just one strain subdivision.
- The length of the real working array must be set to a minimum of 3 if “Set in user coding” is not selected. The length of the integer working array may be set to zero, since no integer history variables are required.
- The nonsymmetric sparse solver will be used when the nonsymmetric stiffness matrix flag and the nonsymmetric sparse solver are selected.

This page intentionally left blank

4. Contact conditions

4.1 Introduction

4.1.1 General contact

- Contact conditions can be specified in ADINA to model contact involving solid elements (2-D and 3-D solids) and/or structural elements (truss, beam, iso-beam or axisymmetric shell, plate, shell and pipe elements).
- Very general contact conditions are assumed:
 - ▶ The points of contact are assumed not known a priori.
 - ▶ Friction can be modeled according to various friction laws.
 - ▶ Repeated contact and separation between multiple bodies is permitted in any sequence.
 - ▶ Self-contact and double-sided contact are permitted.
 - ▶ Node-to-segment or node-to-node contact can be assumed.
 - ▶ A small displacement contact feature is available.
- Some of the contact algorithms used in ADINA are described in the following references:

ref. KJB
Section 6.7

- ref. Bathe, K.J. and Chaudhary, A., "A Solution Method for Planar and Axisymmetric Contact Problems," *Int. J. Num. Meth. in Eng.*, Vol. 21, pp. 65-88, 1985.
- ref. Eterovic, A. and Bathe, K.J., "On the Treatment of Inequality Constraints Arising From Contact Conditions in Finite Element Analysis," *J. Computers & Structures*, Vol. 40, No. 2, pp. 203-209, July 1991.
- ref. Pantuso, D., Bathe, K.J. and Bouzinov, P.A. "A Finite Element Procedure for the Analysis of Thermo-mechanical Solids in Contact," *J. Computers & Structures*, Vol. 75, No. 6, pp. 551-573, May 2000.

- Contact in ADINA is modeled using contact groups, contact surfaces, contact segments and contact pairs, as explained in much greater detail below.
- Note that contact renders the analysis nonlinear even when no nonlinear element groups are defined.

4.1.2 Rigid-target contact

ADINA includes a special contact algorithm for modeling problems in which some of the surfaces are rigid. This “rigid-target algorithm” is described in Section 4.29.

Note that the general contact algorithms described below can also be used when some of the surfaces are rigid. However the rigid-target algorithm includes a drawbead feature that is not available in the general contact algorithms.

4.1.3 Cat contact versions

For the general contact algorithms, there are two contact versions, labeled “Cat” contact versions. (Cat stands for Contactor and Target).

Cat 1 contact: Cat 1 contact represents ADINA developments through program version 9.4.

Cat 2 contact: Cat 2 contact represents the ADINA developments starting from program version 9.5.

As will be seen, many modeling features are available in both Cat 1 and Cat 2 contact. However some modeling features are available only in Cat 1 contact and other modeling features are available only in Cat 2 contact.

Our long-term goal is to make all best practice modeling features available in Cat 2 contact. However, in the short term, there are some useful modeling features that are available only in Cat 1 contact, for example double-sided contact.

Throughout this chapter, the differences in the modeling features between Cat 1 and Cat 2 contact are fully described.

Table 4.1 gives a complete list of features, described using command-line input, and specifies the Cat versions that support those features.

Table 4-1: Features in Cat 1 and Cat 2 contact

Feature	Cat 1	Cat 2
Rigid-target method: MASTER CONTACT-ALGORITHM=RIGID-TARGET CONTACT-CONTROL CONTACT-ALGORITHM=RIGID-TARGET CGROUP ALGORITHM=RIGID-TARGET	Supported in Cat 1	Not supported in Cat 2
Constraint function method: MASTER CONTACT-ALGORITHM= CONSTRAINT-FUNCTION CONTACT-CONTROL CONTACT-ALGORITHM= CONSTRAINT-FUNCTION CGROUP ALGORITHM=CONSTRAINT-FUNCTION	Supported in Cat 1	Supported in Cat 2
Default explicit contact: CONTROL-CONTROL XCONT-ALGORITHM=DEFAULT	Supported in Cat 1, uses kinematic constraint for the central difference method and uses penalty for the Noh- Bathe method	Supported in Cat 2, uses penalty method
Kinematic constraint explicit contact: CONTROL-CONTROL XCONT-ALGORITHM=KINEMATIC- CONSTRAINT CGROUP XALGORITHM=KINEMATIC- CONSTRAINT	Supported in Cat 1	Not supported in Cat 2
Penalty explicit contact: CONTROL-CONTROL XCONT-ALGORITHM=PENALTY CGROUP XALGORITHM=PENALTY	Supported in Cat 1	Supported in Cat 2
Explicit rigid target contact contact: CONTROL-CONTROL XCONT-ALGORITHM= EXPLICIT-RIGID-TARGET CGROUP XALGORITHM= EXPLICIT-RIGID-TARGET	Supported in Cat 1	Not supported in Cat 2

Table 4-1: Features in Cat 1 and Cat 2 contact, continued

Feature	Cat 1	Cat 2
Large displacement contact: CONTACT-CONTROL DISPLACEMENT=LARGE CGROUP DISPLACEMENT=LARGE	Supported in Cat 1	Supported in Cat 2
Small displacement contact: CONTACT-CONTROL DISPLACEMENT=SMALL CGROUP DISPLACEMENT=SMALL	Supported in Cat 1, but always rebuilds contact constraints on restart	Supported in Cat 2, doesn't rebuild contact constraints on restart. It is possible to restart from small displacement to large displacement contact, and vice versa.
Small displacement contact with rebuild on restart: CONTACT-CONTROL DISPLACEMENT=SMALL-REBUILD CGROUP DISPLACEMENT=SMALL-REBUILD	Supported in Cat 1	Supported in Cat 2. It is possible to restart from small displacement to large displacement contact, and vice versa.
Old contact surfaces: CONTACT-CONTROL CSTYPE=OLD	Supported in Cat 1	Not supported in Cat 2
Input contact surface orientations CONTACT-CONTROL ORIENTATION=INPUT	Supported in Cat 1	Supported in Cat 2
Automatic contact surface orientations CONTACT-CONTROL ORIENTATION=AUTOMATIC	Supported in Cat 1	Supported in Cat 2
Continuous normal calculation CONTACT-CONTROL CNCALC CGROUP CNCALC	Ignored in Cat 1	Supported in Cat 2
Continuous normal maximum angle CONTACT-CONTROL MAXCNANGLE CGROUP MAXCNANGLE	Ignored in Cat 1	Supported in Cat 2
CNORMAL specifications: CONTACT-CONTROL CNORMAL CGROUP CNORMAL CONTACTPAIR CNORMAL	Not supported in Cat 1	Supported in Cat 2
CFRICTION specifications: CONTACT-CONTROL CFRICTION CGROUP CFRICTION CONTACTPAIR CFRICTION	Not supported in Cat 1	Supported in Cat 2
CRELDAMP specifications: CONTACT-CONTROL CRELDAMP CGROUP CRELDAMP CONTACTPAIR CRELDAMP	Not supported in Cat 1	Supported in Cat 2
CABSDAMP specifications: CONTACT-CONTROL CABSDAMP	Not supported in Cat 1	Supported in Cat 2

Table 4-1: Features in Cat 1 and Cat 2 contact, continued

Feature	Cat 1	Cat 2
Absolute contact damping CONTACT-CONTROL, DAMPING DAMP-NORMAL, DAMP-TANGENTIAL	Supported in Cat 1	Supported in Cat 2. In Cat 2, it is preferred to use the CABSDAMP command to define absolute contact damping.
CONTACT-CONTROL NSUPPRESS	Supported in Cat 1	Supported in Cat 2
Tension-consistent feature: CONTACT-CONTROL, TENSION-CONSISTENT=YES	Supported in Cat 1	Supported in Cat 2
V83 friction algorithm: CONTACT-CONTROL, FRICTION-ALGORITHM=V83	Supported in Cat 1	Ignored in Cat 2
Default post-impact corrections: CONTACT-CONTROL POSTIMPACT=DEFAULT	Supported in Cat 1	Ignored in Cat 2
Post-impact corrections: CONTACT-CONTROL POSTIMPACT=YES	Supported in Cat 1	Not supported in Cat 2
“Mod” post-impact corrections: CONTACT-CONTROL POSTIMPACT=MOD	Supported in Cat 1	Not supported in Cat 2
Search distance: CONTACT-CONTROL SEARCHDIST CGROUP SEARCHDIST	Ignored in Cat 1	Supported in Cat 2
Opposite normals search test: CONTACT-CONTROL OPPNTEST CGROUP OPPNTEST	Ignored in Cat 1	Supported in Cat 2
3D-search: CONTACT-CONTROL 3D-SEARCH	Supported in Cat 1	Ignored in Cat 2
Frictionless friction-delay: CONTROL-CONTROL FRIC-DELAY=FRICIONLESS CGROUP FRIC-DELAY=FRICIONLESS Stabilized friction-delay: CONTROL-CONTROL FRIC-DELAY=STABILIZED CGROUP FRIC-DELAY=STABILIZED	Supported in Cat 1	Supported in Cat 2
Patch release CONTACT-CONTROL PATCH-RELEASE=YES CGROUP PATCH-RELEASE=YES	Ignored in Cat 1	Supported in Cat 2

Table 4-1: Features in Cat 1 and Cat 2 contact, continued

Feature	Cat 1	Cat 2
No pressure control: CONTACT-CONTROL PRESSURE-CONTROL=NO V89 pressure control: CONTACT-CONTROL PRESSURE-CONTROL=V89 Pressure control: CONTACT-CONTROL PRESSURE-CONTROL=YES	Supported in Cat 1	Supported in Cat 2
V89 traction calculations: CONTACT-CONTROL TRACT-CALC=V89	Supported in Cat 1	Ignored in Cat 2.
Angle tolerance: CONTACT-CONTROL TRACT-ANGLE	Supported in Cat 1, used only for traction calculation equivalent areas.	Supported in Cat 2, used for all equivalent area calculations.
CGROUP CONTACT2 SUBTYPE	Supported in Cat 1, SUBTYPE=STRESS is the same as SUBTYPE=STRAIN	Supported in Cat 2, SUBTYPE=STRESS allows entry of the contact segment thickness, SUBTYPE=STRAIN does not allow entry of the contact segment thickness.
Forces and tractions CGROUP FORCES TRACTIONS	Supported in Cat 1	Supported in Cat 2
Double-sided contact: CGROUP PENETRATION-ALGORITHM=TWO	Supported in Cat 1	Supported in Cat 2, but using the same coding as in Cat 1
Contact segment thickness: CGROUP CONTACT2 THICKNESS	Ignored in Cat 1	Supported in Cat 2
Offsets: CGROUP OFFSET OFFSET-TYPE=CONSTANT CGROUP OFFSET-TYPE=TRUE CGROUP OFFSET-TYPE=NONE	Supported in Cat 1	Supported in Cat 2
Segment normals: CGROUP CONTINUOUS-NORMAL=NO Continuous normals: CGROUP CONTINUOUS-NORMAL=YES	Supported in Cat 1	Supported in Cat 2
Contact surface extension CGROUP CS-EXTENSION	Supported in Cat 1	Supported in Cat 2
Depth: CGROUP DEPTH	Supported in Cat 1	Supported in Cat 2
Contact nodal area: CGROUP NODAL-AREA	Ignored in Cat 1	Supported in Cat 2

Table 4-1: Features in Cat 1 and Cat 2 contact, continued

Feature	Cat 1	Cat 2
Compliance: CGROUP CFACTOR1	Supported in Cat 1	Supported in Cat 2. But it is preferred to use the CNORMAL command to specify CFACTOR1.
CGROUP EPSN	Supported in Cat 1	Supported in Cat 2, but it is preferred to use the CNORMAL command to specify EPSN.
Global normal penalty stiffness CGROUP XKN-CRIT=GLOBAL User-specified normal penalty stiffness CGROUP XKN-CRIT=USER	Supported in Cat 1	Supported in Cat 2. But it is preferred to use the CNORMAL command to specify these parameters.
Friction: CGROUP FRICTION	Supported in Cat 1	Supported in Cat 2. But it is preferred to use the CFRICTION command to specify friction parameters.
CGROUP EPST	Supported in Cat 1	Supported in Cat 2, but it is preferred to use the CFRICTION commands to specify EPST.
User-supplied friction with the CGROUP USER-FRICTION command CGROUP USER-FRICTION	Supported in Cat 1	Not supported in Cat 2. Use the CFRICTION USER-SUPPLIED command for Cat 2.
Global tangential penalty stiffness CGROUP XKT-CRIT=GLOBAL User-specified tangential penalty stiffness CGROUP XKT-CRIT=USER	Supported in Cat 1	Supported in Cat 2. But it is preferred to use the CFRICTION commands to specify these parameters.
Relative penalty damping CGROUP XDAMP=RELATIVE Absolute penalty damping CGROUP XDAMP=ABSOLUTE	Supported in Cat 1	Supported in Cat 2. But it is preferred to use the CRELDAMP commands to specify these parameters.
Node-node contact: CGROUP NODETONODE=YES DIRECTION	Supported in Cat 1	Supported in Cat 2, but using the same coding as in Cat 1
Creation of gluemesh surfaces: CGROUP GLUEMESH=YES	Supported in Cat 1	Supported in Cat 2
Tied contact: CGROUP TIED=SMALL TIED-OFFSET	Supported in Cat 1	Not supported in Cat 2
Contact group birth-death: CGROUP TBIRTH TDEATH	Supported in Cat 1	Supported in Cat 2

Table 4-1: Features in Cat 1 and Cat 2 contact, continued

Feature	Cat 1	Cat 2
Eliminated (allowed) initial penetrations CGROUP INITIAL-PENETRATION=ELIMINATED, ALLOWED, or PRINT	Supported in Cat 1	Supported in Cat 2
Ignored (discarded) initial penetrations CGROUP INITIAL-PENETRATION=IGNORED or DISCARDED	Supported in Cat 1	Supported in Cat 2
Gap-override initial penetrations CGROUP INITIAL-PENETRATION=GAP- OVERRIDE	Supported in Cat 1	Supported in Cat 2
Gap-override time penetration option CGROUP TIME-PENETRATION=(a real number)	Supported in Cat 1	Supported in Cat 2
Gap-override time penetration option CGROUP TIME-PENETRATION=FIRST-STEP	Same as TIME- PENETRATION=0.0 in Cat 1	Supported in Cat 2
Multiphysics CGROUP HHATTMC, FCTMC, FTTMC, EKTMC, GAP-CONDUCTANCE CONTACTPAIR HHATTMC FCTMC FTTMC EKTMC, GAP-CONDUCTANCE	Supported in Cat 1	Supported in Cat 2
Consistent stiffness: CGROUP CONSISTENT-STIFF	Supported in Cat 1	Ignored in Cat 2
Friction specified using COULOMB-FRICTION command	Supported in Cat 1	Not supported in Cat 2. Use the CFRICTION LAW1 and CFRICTION LAW2 commands for Cat 2.
Contact compliance in frequency solutions FREQUENCIES CONT-COMPL	Supported in Cat 1	Supported in Cat 2
Thermal coupling using ADINA Structures and ADINA Thermal	Supported in Cat 1	Not supported in Cat 2. Note that this option can only be detected when ADINA / ADINA-T TMC is actually executed.
FSI analysis using small displacement contact	Not supported in Cat 1	Supported in Cat 2

Notes for Table 4.1

1) Features that are supported in only one Cat version, and that are not supported or ignored by the other Cat version, have cells with gray backgrounds.

4.1.4 Contact concepts overview

Due to the large number of concepts used for modeling contact, a short overview of each concept is given here, with additional details given in later sections of this chapter.

4.1.4.1 Normal contact response

In its most basic form, contact acts to resist penetration of a *contactor node* into a *target segment*. The direction of penetration is given by the normal of the target segment, and the amount of penetration is given by the *overlap* of the contactor node and the target segment, as measured in the direction of the target segment normal. Frequently we will write the overlap as the negative of the *gap* of the contactor node and target segment.

The normal contact response determines the normal contact traction caused by the gap. For example, the normal contact traction might be proportional to the overlap. The term “contact pressure” is also used for the normal contact traction.

For more information about normal contact, see Section 4.2.

4.1.4.2 Frictional contact

When friction is included, then friction acts to resist the relative tangential motion of the contactor node and target segment. In ADINA, this tangential motion is measured using the *slip velocity* of the contactor node. The slip velocity is the relative velocity of the contactor node with respect to the contacting point on the target segment, as projected onto the the tangential plane of the target segment.

The friction model specifies the frictional tractions caused by the relative tangential velocity and normal contact traction of the contactor node.

For more information about frictional contact, see Section 4.3.

4.1.4.3 Absolute contact damping

Occasionally it is convenient to add normal and tangential grounded viscous dampers to all contactor and target nodes in the model. These grounded viscous dampers are termed “absolute contact damping”. Absolute contact damping can be useful in static problems for stabilizing the model especially when there are insufficient boundary conditions to remove rigid body modes.

For more information about absolute contact damping, see Section 4.4.

4.1.4.4 Relative contact damping

Relative contact damping provides a normal contactor traction due to the relative motion of the contactor node and target segment in the normal direction. This relative motion is measured using the *gap velocity*, which is the time derivative of the gap. Relative contact damping can be used for damping out high-frequency oscillations.

For more information about relative contact damping, see Section 4.5.

4.1.4.5 Multiphysics

In thermo-mechanical-coupling analysis, heat transfer can occur between the contacting surfaces. Heat due to frictional contact can be generated onto the contact surfaces.

For more information about thermo-mechanical-coupling and contact, see Section 4.6.

4.1.4.6 Contact groups

All of the contact surfaces of the model are contained in contact groups.

Many of the feature settings for contact surfaces and contact pairs are made at the contact group level. For example, the type of contact surface (2D or 3D) is made at the contact group level, so all

of the contact surfaces in a group are either 2D or 3D.

For more information about contact groups, see Section 4.7.

4.1.4.7 Contact surfaces

A 2D contact surface is made up of connected 2- or 3-node contact segments. The contact surface can be open or closed. 2D contact surfaces are typically attached to the edges of 2D solid elements.

A 3D contact surface is made up by segments, each containing from 3 to 9 nodes. Usually the segments are connected to each other, but it is not necessary for the segments to be connected. The contact surface can be open or closed. 3D contact surfaces are typically attached to the faces of 3D solid elements, or to shell elements, or to plate elements.

The nodes on a contact surface can be free, fixed, prescribed or constrained. If all of the nodes on a contact surface are fixed, prescribed, or constrained to a node which is prescribed, then the contact surface is termed “rigid”, otherwise the contact surface is flexible.

For *single-sided contact surfaces*, the direction of penetration is specified by the normal direction of the contact segments. For *double-sided contact surfaces*, the penetration can occur in either the normal direction, or in the opposite of the normal direction. Double-sided contact is available for 3D contact only.

In *node-to-node contact*, each node on the contactor surface is paired with a node on the target surface. Each target surface node is given a normal direction and tangential plane.

There is also the option of describing a target contact surface as a smooth analytically described surface. An analytically described surface is termed an “*analytical rigid target*” (ART). The smooth surface representation with ARTs allows for more accurate and more easily converging solutions, especially when friction is present. Most of the applications of ARTs are in metal forming problems.

For more information about contact surfaces see Section 4.8.

4.1.4.8 Contact pairs

A contact pair consists of two contact surfaces within the same contact group, that can come into contact. One of the contact surfaces in the pair is designated the *contactor contact surface* and the other surface is designated the *target contact surface*.

Within a contact pair, the nodes of the contactor contact surface are termed “contactor nodes” and the segments of the target contact surface are termed “target segments”.

It is allowed for a contact surface to be a contactor surface in one pair, and to be a target surface in another pair in the same contact group.

Many of the properties associated with contact are defined at the contact pair level. For example, the friction coefficient can be defined for each contact pair.

For more information about contact pairs, see Section 4.9.

4.1.4.9 Contact interactions

The interaction between contact surfaces is generally given in terms of contactor elements.

A contactor element is the combination of a contactor node and a target segment. The program automatically determines the contactor elements during program execution.

The normal, frictional and relative contact damping responses are all described in terms of contactor elements.

For more information about contactor elements and for features that control contact interactions, see Sections 4.10 to 4.22.

4.1.4.10 Contact output

The contact output is node-based. For each contactor node, the following output is available:

- Status
- Gap
- Normal traction
- Slip velocity
- Tangential traction
- Gap velocity
- Relative damping traction
- Nodal point force
- Relative damping nodal point force

For each target node (contact node not used as a contactor), only the nodal point force is available.

For more information about contact output, see Section 4.23.

4.1.4.11 Contact convergence criteria

In implicit analysis, additional contact convergence criteria are used during the equilibrium iterations.

For more information about contact convergence criteria, see Section 4.24.

4.1.4.12 Frequency analysis

Contact conditions can be considered in frequency analysis. Contacting bodies remain in contact during frequency analysis.

For more information about contact in frequency analysis, see Section 4.25.

4.1.4.13 Modeling recommendations

Modeling contact conditions can be quite challenging. Some recommendations are given in Sections 4.26 to 4.28.

4.2 Normal contact

The normal contact response gives the normal contact traction caused by the gap. For example, the normal contact traction might be proportional to the overlap. The term “contact pressure” is also used for normal contact traction.

The normal contact response is different for implicit and explicit analysis.

4.2.1 Fundamental quantities

The following nomenclature is used:

g is the contact gap

h is the contact overlap, $h = -g$

\mathbf{n} is the target normal.

\mathbf{x}^C is the current location of the contactor node

\mathbf{x}^T is the current location of the contacting point on the target surface

λ_g is the normal contact force due to the gap

p_g is the normal contact pressure due to the gap (contact force per unit length for 2D plane strain contact).

A is the contactor node area (contactor node length for 2D plane strain contact)

The gap is computed using the equation

$$g = -(\mathbf{x}^C - \mathbf{x}^T) \cdot \mathbf{n} \quad (4.2-1)$$

Fig. 4.2-1 shows the gap for 2D and 3D contact surfaces.

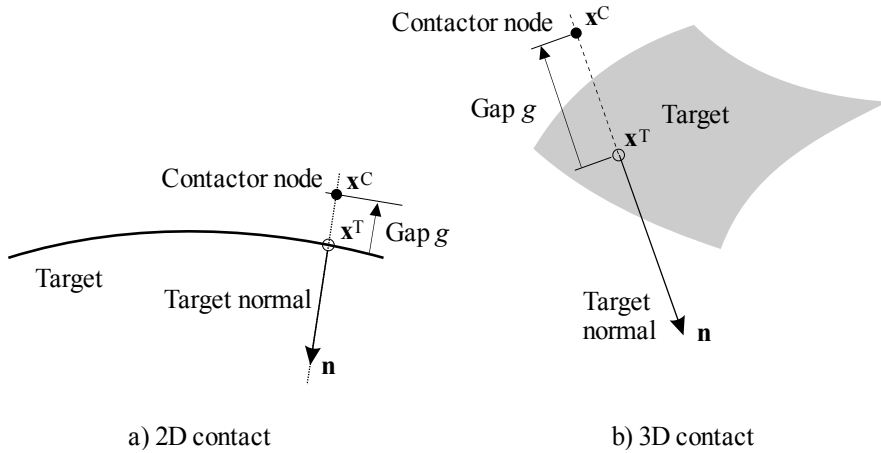


Figure 4.2-1: Gap for 2D and 3D contact surfaces

The normal contact force acting on the contactor node due to the gap is related to the normal contact pressure due to the gap by the formula

$$\lambda_g = A p_g \quad (4.2-2)$$

The gap is always measured in the direction of the target segment normal. The target segment normal is described in detail in Sections 4.8.1.4 and 4.13.

The normal contact force acting on the target surface is equal and opposite to the normal contact force acting on the contactor node.

4.2.2 Normal contact in implicit analysis

4.2.2.1 No-overlap contact

Penetration can be prevented completely using no-overlap normal contact. No-overlap normal contact is specified by setting the compliance factor to zero.

In ideal no-overlap contact, there are no contact forces for positive gap, see Fig. 4.2-2. In ADINA, ideal no-overlap contact is approximated as discussed below.

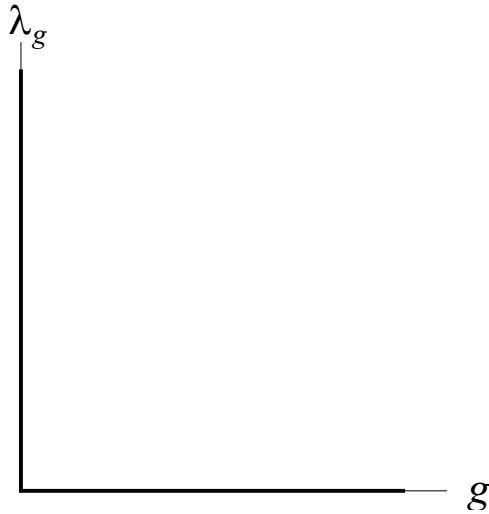


Figure 4.2-2: Ideal no-overlap contact

In Cat 1 contact, the no-overlap contact response is given by the hyperbola

$$\lambda_g = \frac{\varepsilon_N}{g} \quad (4.2-3)$$

in which ε_N is a small user-defined parameter. The no-overlap contact response function is shown in Fig. 4.2-3. The default value of $\varepsilon_N = 1 \times 10^{-12}$ is suitable for most applications and should rarely be modified.

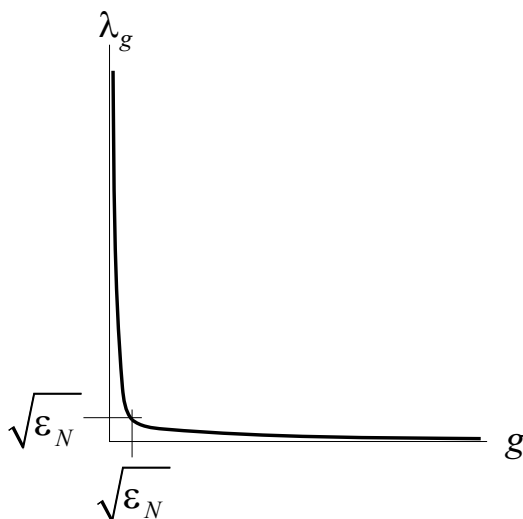


Figure 4.2-3: No-overlap contact in Cat 1 contact

Overlap is prevented in no-overlap contact, however even for positive gap, contact forces are present. Hence this no-overlap contact is not ideal contact. However, we use the term no-overlap contact as a convenient term to distinguish no-overlap contact from compliant contact.

In Cat 2 contact, the no-overlap contact response is given by the curve shown in Fig. 4.2-4. For $g \leq \sqrt{\varepsilon_N}$, the contact response is given by a hyperbola:

$$p_g = \frac{\varepsilon_N}{g} \quad (4.2-4)$$

and for $g > \sqrt{\varepsilon_N}$, the contact response is given by a circular arc that touches the g axis, and then for larger values of g , the contact response is given by a straight line with zero p_g .

In the above, ε_N is a small user-defined parameter that governs the curvature of the contact response near the origin. The default value

of $\varepsilon_N = 1 \times 10^{-12}$ is suitable for most applications and should rarely be modified.

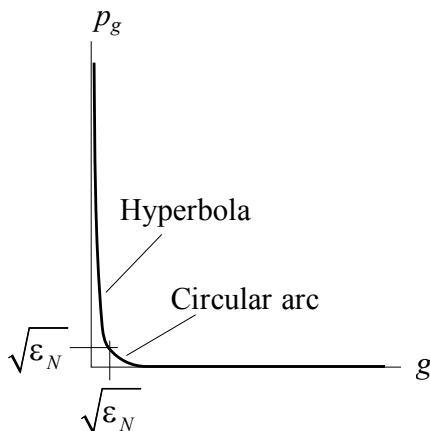


Figure 4.2-4: No-overlap contact in Cat 2 contact

4.2.2.2 Compliant contact with constant compliance factor

The normal contact can be made compliant by setting the compliance factor nonzero.

In Cat 1 contact, the compliant contact response is a smooth curve with asymptotes given by the function

$$\begin{aligned} p_g &= \frac{1}{\hat{c}} \times (-g), \text{ for } g < 0 \\ &= 0, \text{ for } g \geq 0 \end{aligned} \quad (4.2-5)$$

in which \hat{c} is the compliance factor, see Fig. 4.2-5. Although not shown in the equation, ε_N is a small user-defined parameter that governs the curvature of the contact response near the origin.

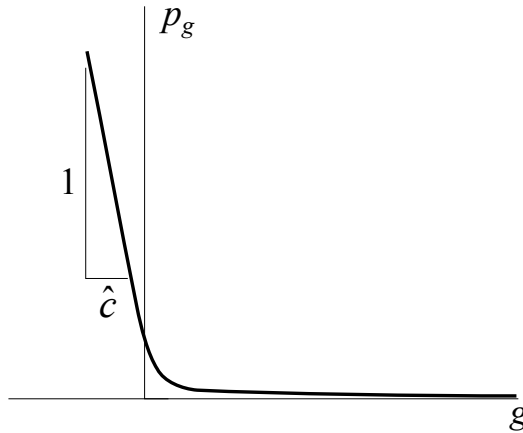


Figure 4.2-5: Compliant contact in Cat 1 contact

In Cat 2 contact, the compliant contact response is a segmented curve given by equation (4.2-5), as shown in Fig. 4.2-6.

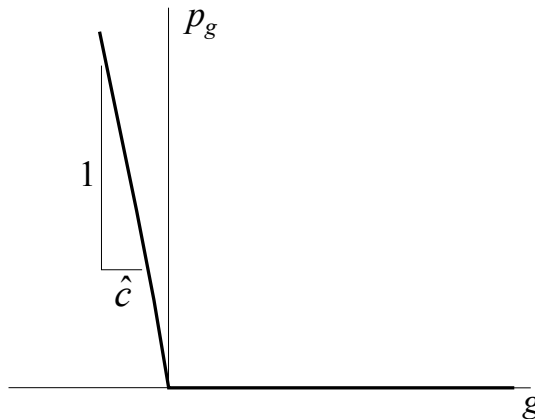


Figure 4.2-6: Compliant contact in Cat 2 contact

For compliant contact, ε_N is not used in Cat 2 contact.

Clearly when the contact is compliant, a contactor node must penetrate the target segment to generate normal contact forces.

The dimensions of \hat{c} are $\frac{[L^3]}{[F]}$.

In Cat 2 contact, \hat{c} can be made time-dependent by specifying a time function. However, the value of \hat{c} must always be positive.

4.2.2.3 Compliant contact with constant compliance factor and backstop

Cat 2 contact also allows compliant contact to be used with a backstop as shown in Fig. 4.2-7.

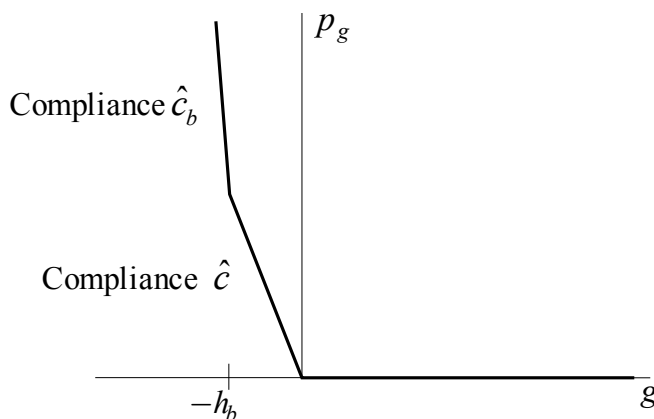


Figure 4.2-7: Compliant contact with backstop in Cat 2 contact

There are two options for specifying the backstop overlap h_b :

Direct: h_b is user-specified.

Scaled: $h_b = \ell_{\min} \times$ a user-specified scale factor, where ℓ_{\min} is the minimum segment length of the contact group.

There are two options for specifying the backstop compliance \hat{c}_b :

Direct: \hat{c}_b is user-specified.

Scaled: $\hat{c}_b = \hat{c} \times$ a user-specified scale factor

The backstop can be used only in implicit analysis.

4.2.2.4 Power law contact pressure function

Cat 2 contact allows the use of a power law contact pressure function.

The power law contact pressure function is shown in Fig. 4.2-8. The following relationship is assumed:

$$x = \frac{g_0 - g}{g_{dist}} \quad (4.2-6)$$

$$p_g = 0 \text{ for } x < 0 \quad (4.2-7a)$$

$$p_g = p_1 x^\alpha \text{ for } x < n + 1 \quad (4.2-7b)$$

$$p_g = p_2 + \left. \frac{dp_g}{dg} \right|_{g_2} (g - g_2) \text{ for } x \geq n + 1 \quad (4.2-7c)$$

in which

$$\left. \frac{dp_g}{dg} \right|_{g_2} = -\frac{p_1}{g_{dist}} \alpha (n + 1)^{\alpha-1} \quad (4.2-8)$$

In the above, g_{dist} , g_1 , p_1 , α and n are user-specified. From these constants,

$$g_0 = g_1 + g_{dist} \quad (4.2-9)$$

$$g_2 = g_1 - n \times g_{dist} \quad (4.2-10)$$

$$p_2 = p_1 (1 + n)^\alpha \quad (4.2-11)$$

are calculated.

The figure is drawn assuming that $\alpha > 1$, however, it is allowed for α to be less than or equal to 1, provided that α is positive.

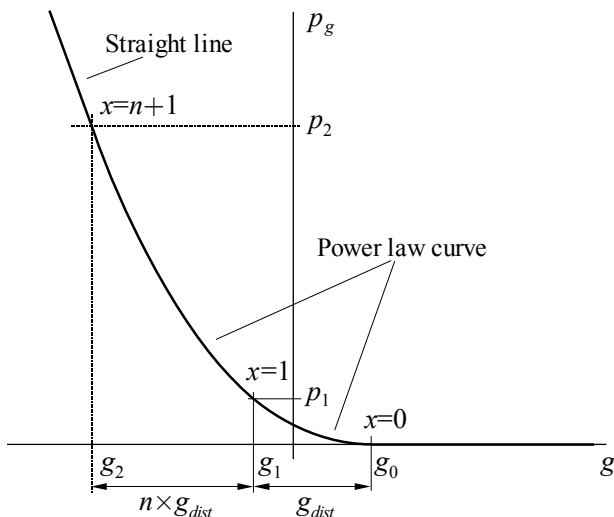


Figure 4.2-8: Power law contact pressure function in Cat 2 contact

The contact pressure from the above relationship can optionally be scaled by a time function.

The so-called Ostrovskii law is

$$\delta = c(p_n)^m \quad (4.2-12)$$

in which δ is the overlap, p_n is the contact pressure and c , m are constants. In our notation, the Ostrovskii law is

$$-g = c(p_g)^m \quad (4.2-13)$$

The Ostrovskii law can be put into power law form:

$$p_g = \left(\frac{-g}{c} \right)^{1/m} \quad (4.2-14)$$

so that the Ostrovskii law can be specified by setting $g_{dist} = c$,

$$g_1 = -c, \quad \alpha = \frac{1}{m}.$$

4.2.2.5 XEXP contact pressure function

Cat 2 contact allows the use of a exponential-like contact pressure function, termed XEXP.

The XEXP contact pressure function is shown in Fig. 4.2-9. The following relationship is assumed:

$$x = \frac{g_0 - g}{g_{dist}} \quad (4.2-15)$$

$$p_g = 0 \text{ for } x < 0 \quad (4.2-16a)$$

$$p_g = \frac{p_1}{\exp(1)-1} \left[x(\exp(x)-1) \right] \text{ for } x < n+1 \quad (4.2-16b)$$

$$p_g = p_2 + \left. \frac{dp_g}{dg} \right|_{g_2} (g - g_2) \text{ for } x \geq n+1 \quad (4.2-16c)$$

in which

$$\left. \frac{dp_g}{dg} \right|_{g_2} = -\frac{p_1}{g_{dist}} \frac{1}{\exp(1)-1} \left[(n+2)\exp(n+1)-1 \right] \quad (4.2-17)$$

In the above, g_{dist} , g_1 , p_1 and n are user-specified. From these constants,

$$g_0 = g_1 + g_{dist} \quad (4.2-18)$$

$$g_2 = g_1 - n \times g_{dist} \quad (4.2-19)$$

$$p_2 = \frac{p_1}{\exp(1)-1} \left[(n+1)(\exp(n+1)-1) \right] \quad (4.2-20)$$

are calculated.

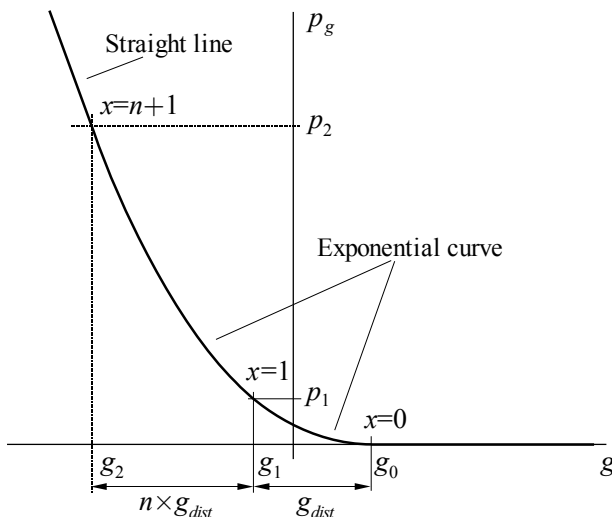


Figure 4.2-9: XEXP contact pressure function in Cat 2 contact

The contact pressure from the above relationship can optionally be scaled by a time function.

4.2.2.6 Multilinear contact pressure function

Cat 2 contact allows the use of a multilinear contact pressure function.

The multilinear contact pressure function is shown in Fig. 4.2-10. A multilinear relationship is assumed based on user-input points g, p_g .

The multilinear curve should ideally have increasing pressure for decreasing gap. The program will accept data in which the multilinear curve is not increasing for increasing gap, but the program may or may not converge. In particular, if the pressure is less than zero, the results may be undefined.

The contact pressure from the above relationship can optionally be scaled by a time function.

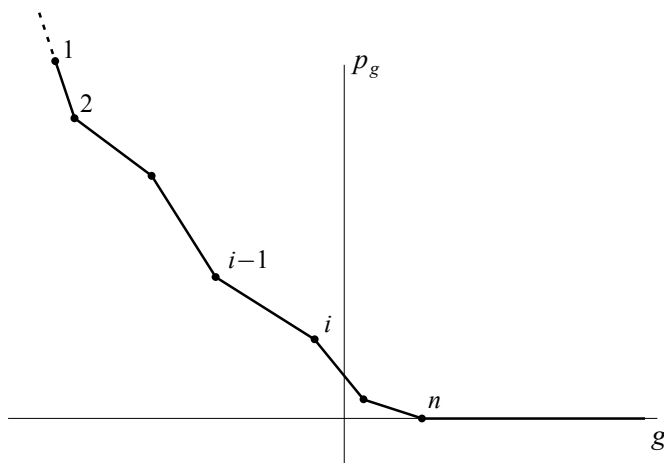


Figure 4.2-10: Multilinear contact pressure function in Cat 2 contact

4.2.3 Normal contact in explicit analysis

The normal contact response in explicit analysis depends on the explicit contact algorithm employed.

4.2.3.1 Kinematic constraint algorithm

The kinematic constraint algorithm is only available in Cat 1 contact. When Cat 1 contact is used with the central difference method, the kinematic constraint algorithm is the default.

A predictor step is first done without applying contact constraints or forces. Then displacements are evaluated and penetration is detected and corrected. The exact correction of displacements requires the solution of a non-diagonal system of equations. Instead, a good approximation is done. In this case, for each penetrating contactor node, a penetration force

$$\mathbf{F}_C^N = M_C \mathbf{a}_C^{N*} = M_C \frac{\delta_N}{\Delta t^2} \mathbf{N} \quad (4.2-21)$$

is calculated. This is the force required to remove the penetration at

the contactor node. However, not all the penetration will be removed by moving the contactor. The target will get some motion depending on its mass relative to the contactor and how many contactor nodes are touching it. So, the \mathbf{F}_C^N force above is projected to the target segment nodes:

$$\mathbf{F}_{T_i}^N = N_i \mathbf{F}_C^N \quad (4.2-22)$$

where N_i is the shape function relating the contactor displacement to that of each target node. Similarly, the mass of the contactor node is projected to the target in the same way:

$$M_{T_i} = N_i M_C \quad (4.2-23)$$

and this mass is added to that of the target node itself. Then the acceleration of the target node is determined as

$$\mathbf{a}_T^N (M_T + \sum M_{T_i}) = \sum \mathbf{F}_{T_i}^N \quad (4.2-24)$$

This correction is then used to update the target displacements. The contactor acceleration is

$$\mathbf{a}_C^N = \mathbf{a}_C^{N*} - \sum \mathbf{a}_T^N N_i \quad (4.2-25)$$

A modification is also required for rigid targets, which are common in contact. The form of the equations in this case depends on whether the rigid target has natural or essential boundary conditions.

4.2.3.2 Penalty algorithm

The penalty method is the default explicit contact algorithm when the Noh-Bathe method is used.

In this algorithm, contact conditions are imposed by penalizing the inter-penetration between contacting surfaces. When a penetration is detected, the contact pressure at the contactor node is

$$p_g = -K_N g \quad (4.2-26)$$

is applied to the contactor node, where K_N is the normal penalty stiffness.

The normal penalty stiffness K_N can be selected by the user, or determined automatically by the program.

When the normal penalty stiffness is determined automatically, the calculation is made based on the masses of the contactor nodes and the time step. The penalty stiffness is selected such that it has a minimal effect on the existing time step.

Note that an unduly small penalty stiffness will lead to excessive penetrations, and an unduly large penalty stiffness will lead to excessive oscillations or unstable explicit time integration.

In Cat 2 contact, there is also the option of directly using the contact pressure functions to obtain the contact pressure at the contactor node.

4.2.4 Compliance energy

The contact compliance energy for a contactor node is defined as

$$E_n = A \int_g^{\infty} p_g dg \quad (4.2-27)$$

The integral $\int_g^{\infty} p_g dg$ is the area under the contact pressure function curve between the current gap g and infinite gap.

The compliance energy is set to zero when no-overlap contact is used.

For 2D plane strain contact, E_n is the compliance energy per unit thickness.

4.2.5 Specification of input

4.2.5.1 CGROUP and CNORMAL commands

In Cat 1 contact, the normal contact response parameters are set with the CGROUP commands:

```
CGROUP ... EPSN, CFACTOR1,  
           XKN-CRIT, XK-NORMAL
```

In Cat 2 contact, the normal contact response parameters of Cat 1 contact can either be set with the CGROUP parameters or with the CNORMAL commands, and the normal contact response parameters added in Cat 2 contact are set with the CNORMAL commands.

```
CNORMAL BASIC  
CNORMAL POWERLAW  
CNORMAL XEXP  
CNORMAL MULTILINEAR
```

For Cat 2 contact, it is preferred to always use the CNORMAL commands. For example, the following command specifications are equivalent for Cat 2 contact.

```
CGROUP CONTACT3 ... CFACTOR1=1E-6
```

and

```
CNORMAL BASIC 1 CFACTOR1=1E-6  
CGROUP CONTACT3 ... CNORMAL=1
```

In particular, if the same normal contact response parameters are to be used for several contact groups, the CNORMAL value can be specified in the CONTACT-CONTROL command to provide a default CNORMAL value for the contact groups:

```
CNORMAL BASIC 1 CFACTOR1=1E-6  
CONTACT-CONTROL ... CNORMAL=1
```

In additional, the CNORMAL value can be specified for individual contact pairs as discussed in Section 4.9.1.

4.2.5.2 Specification of normal contact parameters

Parameter	Cat 1 contact specification	Cat 2 contact specification
ε_N	CGROUP ... EPSN	CGROUP ... EPSN or CNORMAL BASIC ... EPSN
\hat{c}	CGROUP ... CFACTOR1	CGROUP ... CFACTOR1 or CNORMAL BASIC ..., CFACTOR1 CFACT-TF
Backstop selection		CNORMAL BASIC ..., BACKSTOP={NO/YES}
h_b		CNORMAL BASIC ..., BACK-OVERLAP BACK-OVALUE
\hat{c}_b		CNORMAL BASIC ..., BACK-COMPLIANCE BACK-CVALUE
K_N	CGROUP ... , XKN-CRIT XK-NORMAL	CGROUP ... XKN-CRIT XK-NORMAL or CNORMAL BASIC ... XKN-CRIT XK-NORMAL
Multilinear contact pressure function		CNORMAL MULTILINEAR
Power-law contact pressure function		CNORMAL POWERLAW
XEXP contact pressure function		CNORMAL XEXP

4.3 Frictional contact

When friction is included, there is a frictional force acting between the contactor node and the target segment.

4.3.1 Coordinate systems

The coordinate system shown in Fig. 4.3-1 is used for frictional contact. The normal direction is the same as is used in normal contact. The tangential directions, together with the normal direction, form a right-handed orthonormal coordinate system.

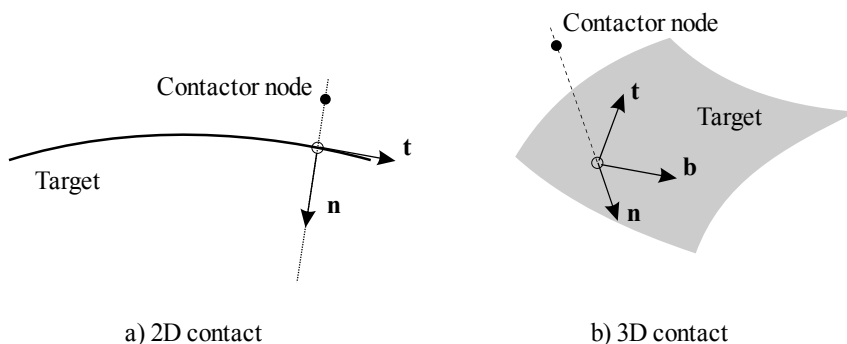


Figure 4.3-1: Coordinate systems used in frictional contact

In 2D contact, the t direction always lies in the 2D plane and the b direction always points out of the 2D plane. When there is contact slip loading, it is possible for there to be a frictional force in the b direction.

4.3.2 Slip velocity

In friction, the slip velocity is used to determine whether sticking or slipping conditions are occurring.

The slip velocity at a contactor node $\dot{\mathbf{u}}$ is defined as the velocity of the contactor node $\dot{\mathbf{u}}^C$ relative to the velocity of the contacting point of the target surface $\dot{\mathbf{u}}^T$, projected onto the tangential plane.

$$\dot{\mathbf{u}} = (\dot{\mathbf{u}}^C - \dot{\mathbf{u}}^T) - \left((\dot{\mathbf{u}}^C - \dot{\mathbf{u}}^T) \cdot \mathbf{n} \right) \mathbf{n} \quad (4.3-1)$$

This definition is opposite to the definition used in ref KJB equation (6.310) (in which the slip velocity is defined as the velocity of the contacting point of the target surface relative to the contactor node, projected onto the tangential plane).

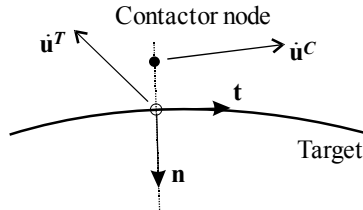


Figure 4.3-2: Slip velocity in 2D contact

If contact slip velocity loads are used, then the velocities of the contactor node and contacting point of the target surface include the slip velocity loads, see the discussion in Section 5.11 for more information about slip velocity loads.

4.3.3 Friction tractions

The direction of the friction traction acting onto the contactor node always points oppositely to the direction of the slip velocity.

The magnitude of the friction traction is denoted T_f . T_f is a function of the sliding friction traction T_{fslide} and the magnitude of the slip velocity \dot{u} . The dependence of T_f on T_{fslide} and the slip velocity magnitude depends on the numerical algorithms employed, and this dependence is discussed in Section 4.19.

In ideal friction, there is a friction traction T_f , even without any slip velocity. However we always assume that the friction traction is zero when the slip velocity is zero.

4.3.4 Friction forces

The magnitude of the friction force τ_f acting onto the contactor node is equal to the magnitude of the friction traction multiplied by the contactor node area:

$$\tau = T_f A \quad (4.3-2)$$

The friction force acting onto the target surface is equal and opposite to the friction force acting onto the contactor node.

The sliding friction force is written τ_{fslide} .

4.3.5 Hard-coded friction models

All of the friction models give the value of the sliding friction traction T_{fslide} in terms of the gap pressure p_g .

4.3.5.1 Coulomb friction

In the Coulomb friction model,

$$T_{fslide} = \mu p_g \quad (4.3-3)$$

where μ is the coefficient of friction.

4.3.5.2 Model 1 friction

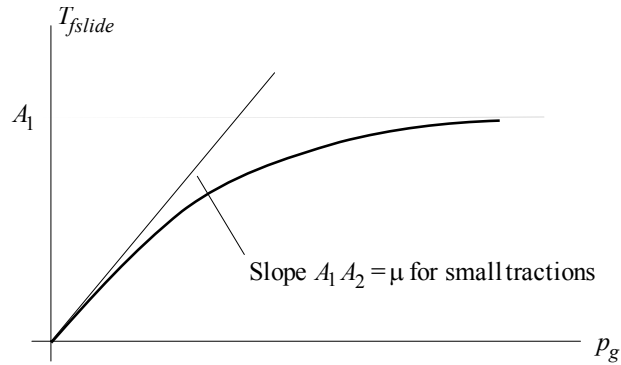
In Model 1 friction,

$$T_{fslide} = A_1 (1 - \exp(-A_2 p_g)) \quad (4.3-4)$$

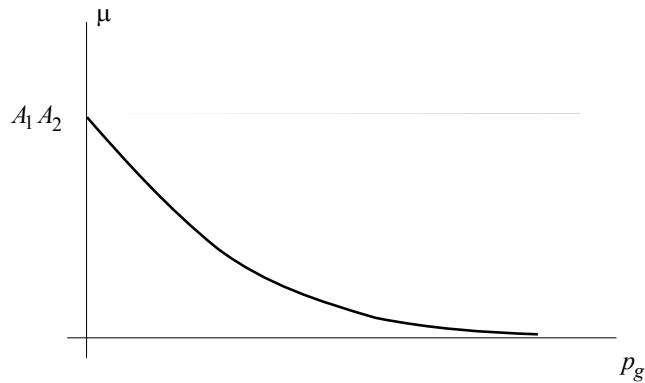
where A_1 , A_2 are specified constants. The instantaneous coefficient of friction is

$$\mu = \frac{T_{fslide}}{p_g} = \frac{A_1 (1 - \exp(-A_2 p_g))}{p_g} \quad (4.3-5)$$

Model 1 friction is graphically shown in Fig. 4.3-3.



a) Sliding friction traction vs gap pressure



b) Coefficient of friction vs gap pressure

Figure 4.3-3: Frictional contact using model 1

4.3.5.3 Model 2 friction

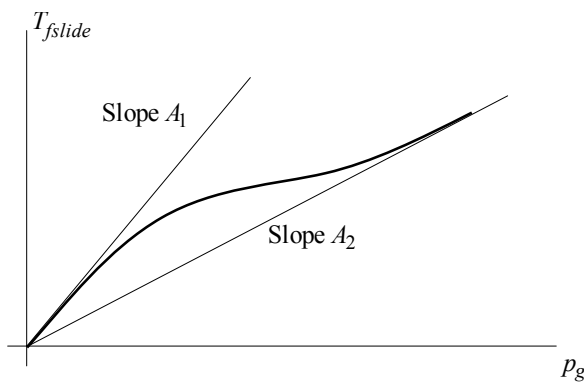
In Model 2 friction,

$$T_{fslide} = [A_2 + (A_1 - A_2) \exp(-A_3 p_g)] p_g \quad (4.3-6)$$

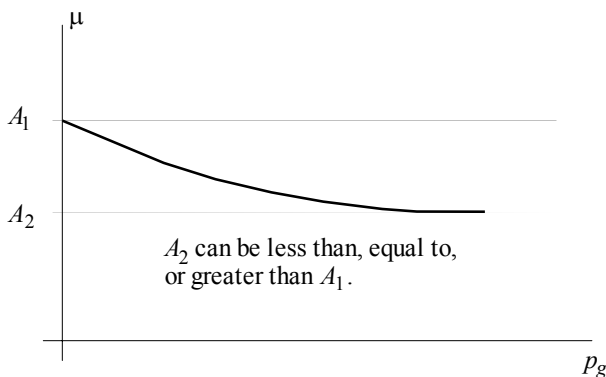
where A_1 , A_2 , A_3 are specified constants. The instantaneous coefficient of friction is

$$\mu = \frac{T_{fslide}}{p_g} = A_2 + (A_1 - A_2) \exp(-A_3 p_g) \quad (4.3-7)$$

Model 2 friction is graphically shown in Fig. 4.3-4.



a) Sliding friction traction vs gap pressure



b) Coefficient of friction vs gap pressure

Figure 4.3-4: Frictional contact using model 2

4.3.6 User-supplied friction models

User supplied friction models are provided in subroutine FUSER for Cat 1 contact and in subroutine CAT2_FUSER for Cat 2 contact. These subroutines can be found in file ovlusr.f.

In Cat 1 contact, the user-supplied friction models are not available in explicit analysis.

The subroutines FUSER and CAT2_FUSER accept two user-specified arrays of constants, an integer array IFPARM and a real array RFPARM. In the current implementation, the first integer constant IFPARM(1) is used to select a friction law. The real constants starting with RFPARM(1) are used as parameters for the specified friction law.

Other inputs to FUSER and CAT_FUSER are described in the comments at the top of the subroutines.

Subroutines FUSER returns the coefficient of friction SCF, an error flag IERR, and an error string ERMES.

Subroutine CAT2_FUSER returns the sliding frictional traction TFS, the partial derivative of TFS with respect to contact pressure DTFS_DP, an error flag IERR and an error string ERMES.

Several friction models are already implemented in subroutines FUSER and CAT2_FUSER. In the descriptions below,
 $A_i = \text{RFPARM}(i)$

In all cases, the sliding frictional traction T_{slide} is

$$T_{slide} = \mu p_g \quad (4.3-8)$$

in which the definition of μ is different for each friction model.

Constant coefficient of friction (IFPARM(1) = 1):

$$\mu = A_1 \quad (4.3-9)$$

Model 1 friction (IFPARM(1)=2):

$$\mu = \frac{A_1 (1 - \exp(-A_2 p_g))}{p_g} \quad (4.3-10)$$

Model 2 friction (IFPARM(1)=3):

$$\mu = \frac{T_{fslide}}{p_g} = A_2 + (A_1 - A_2) \exp(-A_3 p_g) \quad (4.3-11)$$

Different static and dynamic friction coefficients (IFPARM(1)=4)

$$\mu = \begin{cases} A_1 & \text{if } \dot{u} \leq A_3 \\ A_2 & \text{if } \dot{u} > A_3 \end{cases} \quad (4.3-12)$$

Friction coefficient varying with sliding velocity (IFPARM(1)=5)

$$\mu = \begin{cases} A_1 + \frac{\dot{u}}{A_2} (A_3 - A_1) & \text{if } \dot{u} < A_2 \\ A_3 & \text{if } \dot{u} > A_2 \end{cases} \quad (4.3-13)$$

Anisotropic friction model (IFPARM(1)=6)

$$\mu = \begin{cases} \sqrt{(A_1 u_x)^2 + (A_2 u_y)^2 + (A_3 u_z)^2} & \text{if } \dot{u} > A_5 \\ A_4 & \text{if } \dot{u} \leq A_5 \end{cases} \quad (4.3-14)$$

where u_x, u_y, u_z are the x, y and z components of the slip velocity.

Friction coefficient varying with consistent contact force (IFPARM(1)=7)

$$\mu = A_1 + A_2 F_n, \quad 0 \leq \mu \leq 1 \quad (4.3-15)$$

Time varying friction model (IFPARM(1)=8)

$$\mu = \begin{cases} A_1 + \frac{t}{A_2}(A_3 - A_1) & \text{if } t < A_2 \\ A_3 & \text{if } t > A_2 \end{cases} \quad (4.3-16)$$

Coordinate-dependent friction model (IFPARM(1)=9)

$$\mu = A_1 + A_3x + A_4y + A_5z, \quad 0 \leq \mu \leq A_2 \quad (4.3-17)$$

Temperature-dependent friction model (IFPARM(1)=10)

$$\mu = \begin{cases} A_2 & \text{if } \theta < A_1 \\ A_2 \frac{A_3 - \theta}{A_3 - A_1} + A_4 \frac{\theta - A_1}{A_3 - A_1} & \text{if } \theta < A_3 \\ A_4 & \text{if } \theta \geq A_3 \end{cases} \quad (4.3-18)$$

Friction model 1b (IFPARM(1)=12)

$$\mu = \frac{1 - \exp(-A_2 F_n)}{F_n / A_1} \quad (4.3-19)$$

Friction model 2b (IFPARM(1)=13)

$$\mu = A_2 + (A_2 - A_1) \exp(-A_3 F_n) \quad (4.3-20)$$

4.3.7 Frictional contact in explicit analysis

4.3.7.1 Kinematic constraint method

Frictional contact in the kinematic constraint method is similar to normal contact in the kinematic constraint method.

A correction force is calculated

$$\mathbf{F}_C^{T*} = M_C \frac{\mathbf{v}_T}{\Delta t} \quad (4.3-21)$$

where \mathbf{v}_T is the tangential sliding velocity. However, the correction force cannot exceed the limit force based on the normal force and the coefficient of friction

$$\mathbf{F}_C^T = \min(\mu \mathbf{F}_C^N, \mathbf{F}_C^{T*}) \quad (4.3-22)$$

The rest of the procedure is very similar to the case of normal contact. The form of the equations is different if there is damping, and is also different if the previous and current time steps are not the same.

4.3.7.2 Penalty method

The same sliding frictional traction is used as in implicit analysis. The frictional traction for a given sliding velocity is described in Section 4.21.3.

4.3.8 Frictional contact dissipation energy

The frictional contact dissipation energy for a contactor node is calculated using the incremental formula

$${}^{t+\Delta t}E_f = {}^tE_f - \left[\frac{1}{2} ({}^tF_f + {}^{t+\Delta t}F_f) \dot{u} \Delta t \right] \quad (4.3-23)$$

where the negative sign is used so that the dissipation energy is positive.

For 2D planar contact, ${}^{t+\Delta t}E_f$ is the frictional contact dissipation energy per unit thickness.

4.3.9 Frictional heat generation

The heat generation resulting from frictional contact can be accounted for in a thermomechanically coupled analysis. The user can select the fraction of this heat that is lost, as well as the fractions going into the contactor and target surfaces. See Section 4.6.2 for details.

4.3.10 Specification of input

4.3.10.1 CGROUP, CONTACTPAIR and CFRICTION commands

In Cat 1 contact, the frictional contact response parameters are set with the CGROUP and CONTACTPAIR commands:

```
CGROUP ...    FRICTION
CONTACTPAIR ... FRICTION
```

In Cat 2 contact, the frictional contact response parameters of Cat 1 contact can either be set with the CGROUP and CONTACTPAIR parameters or with the CFRICTION commands, and the frictional contact response parameters added in Cat 2 contact are set with the CFRICTION commands.

```
CFRICTION COULOMB
CFRICTION LAW1
CFRICTION LAW2
CFRICTION USER-SUPPLIED
```

For Cat 2 contact, it is preferred to always use the CFRICTION commands. For example, the following command specifications are equivalent for Cat 2 contact.

```
CGROUP CONTACT3 ... FRICTION=0.2
```

and

```
CFRICTION COULOMB 1 SCF=0.2
CGROUP CONTACT3 ... CFRICTION=1
```


In particular, if the same frictional contact response parameters are to be used for several contact groups, the CFRICTION value can be specified in the CONTACT-CONTROL command to provide a default CFRICTION value for the contact groups:

CFRICTION COULOMB 1 SCF=0.2
CONTACT-CONTROL ... CFRICTION=1

In addition, the CFRICTION value can be specified for individual contact pairs as discussed in Section 4.9.2.

4.3.10.2 Specification of frictional contact parameters

Coulomb friction

Parameter	Cat 1 contact specification	Cat 2 contact specification
μ	CGROUP ... FRICTION	CGROUP ... FRICTION or CFRICTION COULOMB ... SCF

Friction law 1

Parameter	Cat 1 contact specification	Cat 2 contact specification
A_1, A_2	COULOMB-FRICTION $a_1 a_2$	CFRICTION LAW1 ... A_1, A_2

Friction law 2

Parameter	Cat 1 contact specification	Cat 2 contact specification
A_1, A_2, A_3	COULOMB-FRICTION $a_1 a_2, a_3$	CFRICTION LAW2 ... A_1, A_2, A_3

User-supplied friction

Parameter	Cat 1 contact specification	Cat 2 contact specification
Integer and real user-supplied parameters	USER-FRICTION integer _i real _i	CFRICTION USER-SUPPLIED integer _i real _i

4.4 Absolute contact damping

When absolute contact damping is used, normal and tangential grounded viscous dampers are added to all contactor and target nodes in the model.

Absolute contact damping can be useful in static problems for stabilizing the model especially when there are insufficient boundary conditions to remove rigid body modes. Absolute contact damping can also be useful in dynamic analysis to dampen out high frequency numerical oscillations.

Absolute contact damping can be set to act only at the initial time step, or to be constant throughout the analysis.

Using the initial absolute contact damping option, the damping is active at the beginning of the first time step, and is reduced gradually (between equilibrium iterations) until it fully dies out by the end of the first time step. Thus the solution at the first time step is free of any absolute contact damping. Note that if contact is not established and nothing else stabilizes the model, the program will not converge.

When absolute contact damping remains active throughout the analysis, the program outputs the sum of all damping forces in the output file, and the user must check that these forces are significantly smaller than the sum of the reaction forces (also written to the output file).

The absolute contact damping forces do not depend on the state of contact.

In Cat 2 contact, the damping constants can be functions of time.

Absolute contact damping can either be force-based or traction-based.

4.4.1 Force-based absolute contact damping

At each contact node (contactor node or target node),

$$\mathbf{F}_{absdamp} = C_N \dot{\mathbf{u}}_N + C_T \dot{\mathbf{u}}_T \quad (4.4-1)$$

in which $\dot{\mathbf{u}}_N$ is the absolute velocity of the contact node in the normal direction, $\dot{\mathbf{u}}_T$ is the absolute velocity of the contact node in the tangential plane and C_N, C_T are damping constants. When $C_T = C_N$, the damping force is isotropic.

The dimensions of C_N and C_T are $\frac{[F] - [T]}{[L]}$.

4.4.2 Traction-based absolute contact damping (not available in Cat 1 contact)

At each contact node (contactor node or target node),

$$\mathbf{T}_{absdamp} = C_N \dot{\mathbf{u}}_N + C_T \dot{\mathbf{u}}_T \quad (4.4-2)$$

$$\mathbf{F}_{absdamp} = A \mathbf{T}_{absdamp} \quad (4.4-3)$$

The dimensions of C_N and C_T are $\frac{[F] - [T]}{[L^3]}$.

4.4.3 Specification of absolute contact damping

Absolute contact damping uses the same parameters for each contact group in the model. It is not possible to use different absolute contact damping parameters for different contact groups.

For Cat 1 contact, force-based absolute contact damping is specified using the DAMPING, DAMP-NORMAL and DAMP-TANGENTIAL parameters of the CONTACT-CONTROL command.

For Cat 2 contact, the same parameters can be used to specify force-based contact damping. However, the CABSDAMP

command can also be used in Cat 2 contact to specify either force-based or traction-based contact damping.

Absolute contact damping is not activated by default.

See Section 4.27 for modeling hints on using absolute contact damping to handle improperly supported structures and how to choose the damping constants.

4.4.4 Absolute contact damping dissipation energy

The absolute contact damping dissipation energy for a contact node is calculated using the incremental formula

$${}^{t+\Delta t}E_{absdamp} = {}^tE_{absdamp} - \left[\frac{1}{2} \left({}^t\mathbf{F}_{absdamp} + {}^{t+\Delta t}\mathbf{F}_{absdamp} \right) \cdot \dot{\mathbf{u}}\Delta t \right] \quad (4.4-4)$$

where the negative sign is used so that the dissipation energy is positive.

4.4.5 Specification of input

4.4.5.1 CONTACT-CONTROL and CABSDAMP commands

In Cat 1 contact, the absolute contact damping parameters are set with the CONTACT-CONTROL command:

```
CONTACT-CONTROL ... DAMPING  
DAMP-NORMAL DAMP-TANGENTIAL
```

In Cat 2 contact, the absolute contact damping parameters of Cat 1 contact can either be set with the CONTACT-CONTROL parameters or with the CABSDAMP command, and the absolute contact damping parameters added in Cat 2 contact are set with the CABSDAMP command.

```
CABSDAMP BASIC
```

For Cat 2 contact, it is preferred to always use the CABSDAMP command. For example, the following command specifications are equivalent for Cat 2 contact.

```
CONTACT-CONTROL ... DAMPING=INITIAL,
DAMP-NORMAL=0.1 DAMP-TANGENTIAL=0.2
```

and

```
CABSDAMP BASIC ... TIMESTEP=INITIAL,
DAMP-NORMAL=0.1 DAMP-TANGENTIAL=0.2
CONTACT-CONTROL ... CABSDAMP=1
```

4.4.5.2 Specification of absolute contact damping parameters

Parameter	Cat 1 contact specification	Cat 2 contact specification
Initial/constant damping	CONTACT-CONTROL DAMPING= {INITIAL/CONSTANT}	CONTACT-CONTROL DAMPING= {INITIAL/CONSTANT} or CABSDAMP BASIC TIMESTEP={INITIAL/ALL}
Force-based/traction based damping	Always force-based	CABSDAMP BASIC TYPE={FORCE/TRACTION}
C_N	CONTACT-CONTROL DAMP-NORMAL	CONTACT-CONTROL DAMP-NORMAL or CABSDAMP BASIC ... DAMP-NORMAL DAMPN-TF
C_T	CONTACT-CONTROL DAMP-TANGENTIAL	CONTACT-CONTROL DAMP-TANGENTIAL or CABSDAMP BASIC ... DAMP-TANGENTIAL DAMPT-TF

4.5 Relative contact damping

Relative contact damping adds dampers between contactor nodes and target segments. With relative contact damping, the normal contact traction depends on the gap velocity as well as on the gap.

The relative contact damping response is different for implicit and explicit analysis.

4.5.1 Fundamental quantities

The following nomenclature is used:

\dot{g} is the contact gap velocity (positive when the gap is increasing)

$\dot{\mathbf{u}}^C$ is the current velocity of the contactor node

$\dot{\mathbf{u}}^T$ is the current velocity of the contacting point on the target surface

$p_{reldamp}$ is the normal contact pressure due to the gap velocity (relative damping contact pressure)

g_{full} is the value of the gap below which the relative contact damping pressure has its “full” value.

g_{ramp} is the gap interval over which the relative contact damping pressure is scaled from 0 to its full value.

The contact gap velocity is computed using the equation

$$\dot{g} = -(\dot{\mathbf{u}}^C - \dot{\mathbf{u}}^T) \cdot \mathbf{n} \quad (4.5-1)$$

The normal contact force due to the relative contact damping is related to the relative contact damping pressure by the formula

$$\lambda_{reldamp} = Ap_{reldamp} \quad (4.5-2)$$

$\lambda_{reldamp}$ acts onto the contactor node. An equal and opposite force acts onto the target surface.

Fig. 4.5-1 shows the relative contact damping damper, gap velocity and relative contact damping forces.

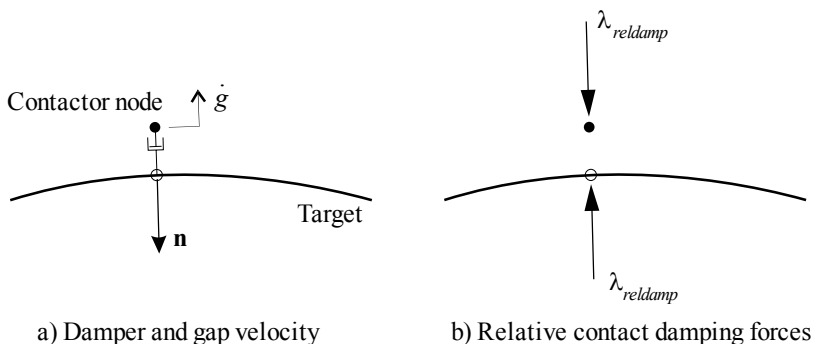


Figure 4.5-1: Relative contact damping

4.5.2 Relative contact damping in implicit analysis

In implicit analysis, relative contact damping is only available in Cat 2 contact.

Relative contact damping is, in addition, only available when the normal contact response is compliant. The no-overlap contact pressure function is not compliant, so relative contact damping cannot be used along with the no-overlap contact pressure function. The power law, XEXP and multilinear contact pressure functions are considered to be compliant, so relative contact damping can be used with these contact pressure functions.

4.5.2.1 Basic relative contact damping

For gap less than g_{full} , the relative contact damping pressure is

$$p_{reldamp} = p_{full} \quad (4.5-3)$$

where

$$\begin{aligned} p_{full} &= C_{nd} |\dot{g}|^\alpha, \dot{g} < 0 \\ &= -C_{nd} |\dot{g}|^\alpha, \dot{g} > 0 \end{aligned} \quad (4.5-4)$$

in which C_{nd} and α are material constants. Fig 4.5-2 shows the dependence of relative damping pressure on the gap velocity, for gap less than g_{full} .

For gap between g_{full} and $g_{full} + g_{ramp}$, the relative contact damping pressure is

$$p_{reldamp} = p_{full} \frac{g_{full} + g_{ramp} - g}{g_{ramp}} \quad (4.5-5)$$

in other words, the relative contact damping pressure is linearly ramped down over the gap interval g_{ramp} from p_{full} to 0.

For gap greater than $g_{full} + g_{ramp}$, the relative contact damping pressure is zero.

Fig 4.5-3 shows the dependence of relative contact damping pressure on the gap.

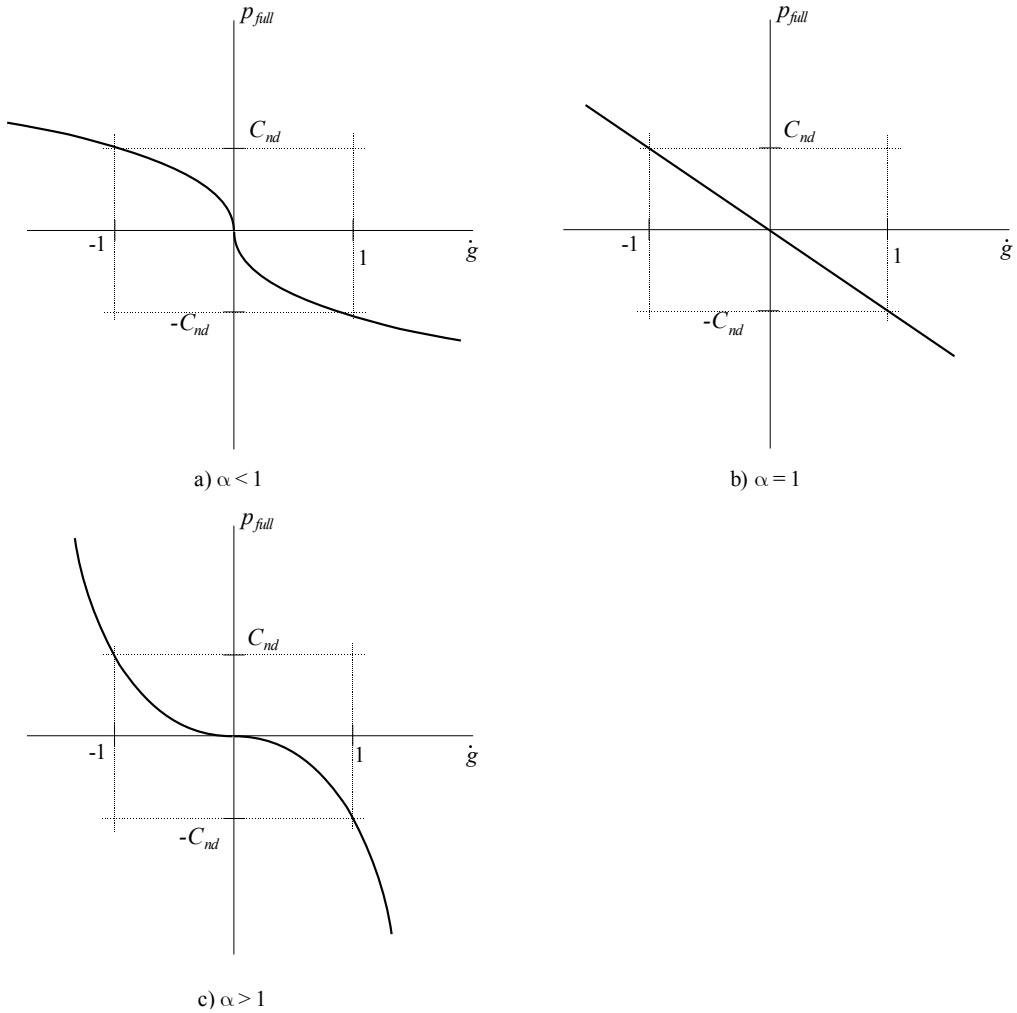


Figure 4.5-2: Dependence of relative contact damping pressure on gap velocity for basic relative contact damping

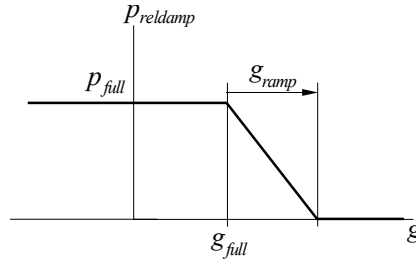


Figure 4.5-3: Dependence of relative contact damping pressure on gap

4.5.2.2 Power law relative contact damping

Fig. 4.5-4 shows power law relative contact damping. The following relationships are used:

$$p_{full} = p_{dp} \left(\frac{\dot{g}}{\dot{g}_p} \right)^{\alpha_p}, \quad \dot{g} > 0 \quad (4.5-6a)$$

$$p_{full} = p_{dn} \left(\frac{\dot{g}}{\dot{g}_n} \right)^{\alpha_n}, \quad \dot{g} < 0 \quad (4.5-6b)$$

with properties p_{dp} , \dot{g}_p , α_p , p_{dn} , \dot{g}_n , α_n .

Fig. 4.5-4 is drawn with α_p , α_n both greater than 1. However, as long as α_p and α_n are both greater than zero, there is no other restriction on α_p and α_n .

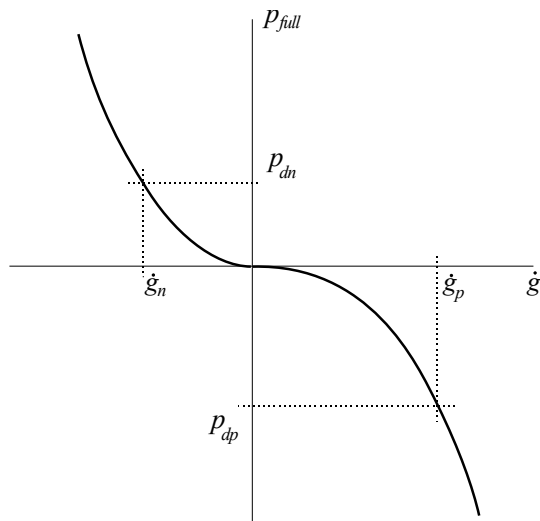


Figure 4.5-4: Dependence of relative contact damping pressure on gap velocity for power law relative contact damping

The relative contact damping pressure can be scaled using time function scaling and gap scaling, as follows:

Time function scaling: The relative contact damping can be scaled by a time function. Two time functions are used, one for positive gap velocity and the other for negative gap velocity.

Gap scaling: For contact gap less than g_{full} , the relative contact damping given by the power law is used without additional gap scaling. For contact gap greater than g_{full} , the relative contact damping pressure is scaled using the gap and g_{ramp} , using the same relationship as for basic relative contact damping, see equation (4.5-5). Two sets of g_{full} , g_{ramp} are used, one set used for positive gap velocity and the other set used for negative gap velocity.

4.5.2.3 Multilinear relative contact damping

Fig. 4.5-5 shows multilinear relative contact damping. The relative contact damping is a multilinear damping law with user-input pairs of \dot{g} , p_d .

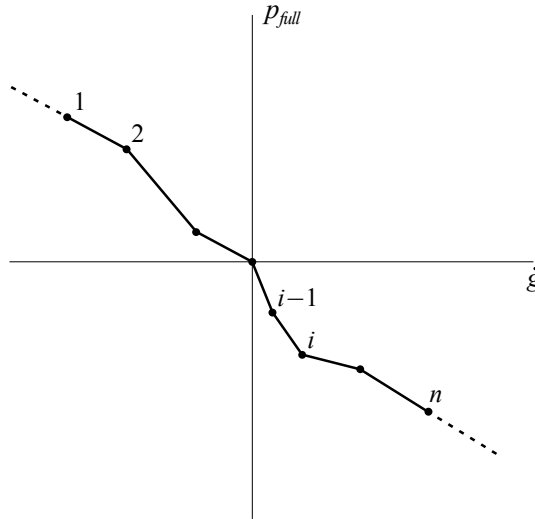


Figure 4.5-5: Dependence of relative contact damping pressure on gap velocity for multilinear relative contact damping

The following restrictions apply to the points (\dot{g}, p_d) :

- a) The point (0,0) must be included.
- b) The sign of p_d must be opposite to the sign of \dot{g} , for all user-input points.
- c) The program will extrapolate for values of \dot{g} outside the range of user-input points.
- d) Each user-input point must have a different value of \dot{g} .
- e) At least two points must be entered.

The multilinear curve need not be monotonically decreasing. There is no limit to the number of points that can be entered.

The relative contact damping pressure can be scaled using time function scaling and gap scaling, similarly to the time function scaling and gap scaling used in power law relative contact damping.

4.5.3 Relative contact damping in explicit analysis

Relative contact damping in explicit analysis is available only when the penalty method is used. Relative contact damping can be used both in Cat 1 and Cat 2 contact.

The formula employed for Cat 1 contact is

$$\begin{aligned} p_{reldamp} &= -K_D \dot{g}, \quad \dot{g} < 0 \\ &= 0, \quad \dot{g} > 0 \end{aligned} \quad (4.5-7)$$

where K_D is a penalty rate damping constant.

The penalty rate damping constant K_D can be explicitly selected by the user, or determined by the program as a ratio of critical damping for the contact node.

For Cat 2 contact, either the Cat 1 formula can be used, or the formulas for implicit analysis can be used.

4.5.4 Relative contact damping dissipation energy

The relative contact damping dissipation energy for a contactor node is calculated using the incremental formula

$${}^{t+\Delta t}E_{reldamp} = {}^tE_{reldamp} - \left[\frac{1}{2} \left({}^t\mathbf{F}_{reldamp} + {}^{t+\Delta t}\mathbf{F}_{reldamp} \right) \cdot \dot{\mathbf{u}}\Delta t \right] \quad (4.5-8)$$

where the negative sign is used so that the dissipation energy is positive.

4.5.5 Specification of input

4.5.5.1 CGROUP and CRELDAMP commands

In Cat 1 contact, the relative contact damping response parameters are set with the CGROUP commands:

```
CGROUP ... XDAMP XNDAMP
```

In Cat 2 contact, the normal contact response parameters of Cat 1 contact can either be set with the CGROUP parameters or with the CRELDAMP command, and the normal contact response parameters added in Cat 2 contact are set with the CRELDAMP commands.

```
CRELDAMP BASIC  
CRELDAMP POWERLAW  
CRELDAMP MULTILINEAR
```

For Cat 2 contact, it is preferred to always use the CRELDAMP commands. For example, the following command specifications are equivalent for Cat 2 contact.

```
CGROUP CONTACT3 ... XNDAMP=0.1
```

and

```
CRELDAMP BASIC 1 XNDAMP=0.1  
CGROUP CONTACT3 ... CRELDAMP=1
```

In particular, if the same relative contact damping response parameters are to be used for several contact groups, the CRELDAMP value can be specified in the CONTACT-CONTROL command to provide a default CRELDAMP value for the contact groups:

```
CRELDAMP BASIC 1 XNDAMP=0.1  
CONTACT-CONTROL ... CRELDAMP=1
```

In addition, the CRELDAMP value can be specified for individual contact pairs as discussed in Section 4.9.3.

4.5.5.2 Specification of relative contact damping parameters

Parameter	Cat 1 contact specification	Cat 2 contact specification
C_{nd}		CRELDAMP BASIC ... CND
α		CRELDAMP BASIC ... ALPHA
g_{full}		CRELDAMP BASIC ... GAPFULL
g_{ramp}		CRELDAMP BASIC ... GAPRAMP
K_D	CGROUP ... XDAMP, XNDAMP	CGROUP ... , XDAMP, XNDAMP or CRELDAMP BASIC ... XDAMP, XNDAMP
Multilinear relative contact damping		CRELDAMP MULTILINEAR
Power law relative contact damping		CRELDAMP POWERLAW

4.6 Multiphysics features

In thermo-mechanical-coupling analysis, the thermal solution is affected by the contact conditions.

Two effects are included:

Heat transfer between contact surfaces

Heat generation due to friction

4.6.1 Heat transfer between contact surfaces

4.6.1.1 Theory

Figure 4.6-1 shows heat transfer between a contactor surface and a target surface. The heat flux per unit area applied to the contactor surface is

$$q^{SC} = \omega \hat{h} (\theta^T - \theta^C) \quad (4.6-1)$$

and the heat flux per unit area applied to the target surface is

$$q^{ST} = \omega \hat{h} (\theta^C - \theta^T) \quad (4.6-2)$$

(Heat flux per unit area is positive when heat is flowing into the surface, and is negative when heat is flowing out of the surface.) The heat flux per unit area applied to the contactor surface is equal and opposite to the heat flux per unit area applied to the target surface.

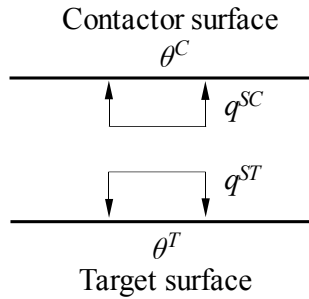


Figure 4.6-1: Heat transfer between contacting surfaces

In the above equations, ω is the gap conductance, \hat{h} is the contact heat transfer coefficient, θ^C is the temperature of the contactor surface and θ^T is the temperature of the target surface. We will describe the gap conductance in more detail below.

The above equations are included in the principle of virtual temperatures as the term

$$\int_{S_c} q^{SC} \delta\theta \, dS + \int_{S_T} q^{ST} \delta\theta \, dS \quad (4.6-3)$$

Since the target contact area is the same as the contactor contact area, and the heat flux per unit area applied to the target surface is equal and opposite to the heat flux applied to the contactor surface, this term is rewritten as an integral over just the contactor surface.

$$\int_{S_c} q^{SC} (\delta\theta^C - \delta\theta^T) \, dS \quad (4.6-4)$$

Because the contactor surface is discretized using contactor nodes, the integral is replaced by a sum:

$$\sum_{\text{Contactor nodes}} q^{SC} A_c (\delta\theta^C - \delta\theta^T) \quad (4.6-5)$$

where A_c is the contactor node area. The physical situation corresponding to equation (4.6-5) is shown in Figure 4.6-2.

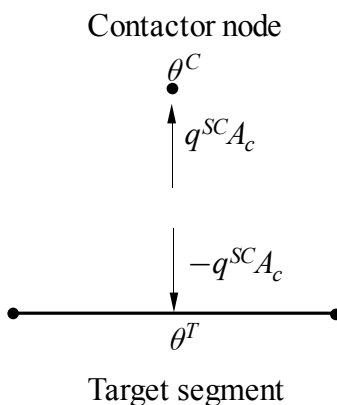


Figure 4.6-2: Heat transfer between contactor node and target segment

Notice that the heat added to the contactor node is equal to and opposite to the heat added to the target segment.

4.6.1.2 Gap conductance options

The gap conductance controls the amount of heat transfer. The gap conductance can be used to simulate conductive (and convective) heat transfer that might occur in the clearance gap before contact. For example, the gap conductance can be used to model the reduction in heat transfer due to surface asperities.

In addition, gap conductance can significantly improve the convergence rate by smoothing the discontinuity in heat conductance at initial contact.

The gap conductance ω has a separate value for each contactor node. Therefore in the discussion below, we consider a single contactor node.

The following gap conductance options are available:

Contact-based gap conductance: The gap conductance is equal to 0 when the contactor node is out of contact and is equal to 1 when the contactor node is in contact, as shown in Figure 4.6-3. This option is the default.

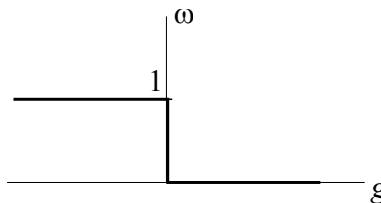


Figure 4.6-3: Contact-based gap conductance

Gap-based gap conductance: The gap conductance ω is a function of the clearance gap g of the contactor node, as shown in Figure 4.6-4. The critical clearance gap g_{\max} must be defined.

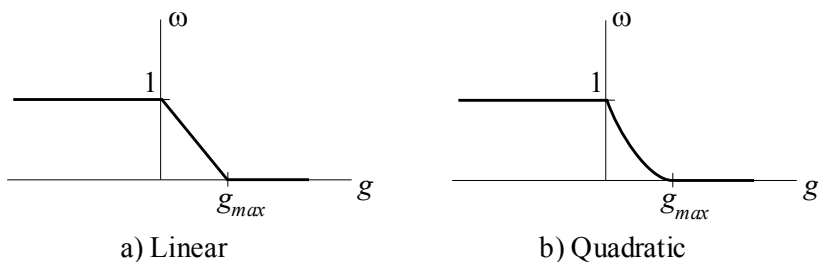


Figure 4.6-4: Gap-based gap conductance

Pressure-based gap conductance: The gap conductance ω is a function of the contact pressure p of the contactor node, as shown in Figure 4.6-5. The critical contact pressure p_{\max} must be defined.

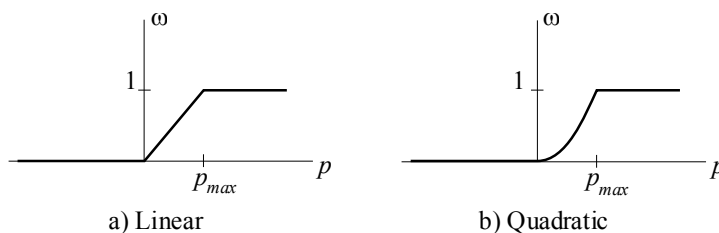


Figure 4.6-5: Pressure-based gap conductance

Gap and pressure-based gap conductance: The gap conductance ω is a function of the clearance gap g , and also a function of the contact pressure p , as shown in Figure 4.6-6.

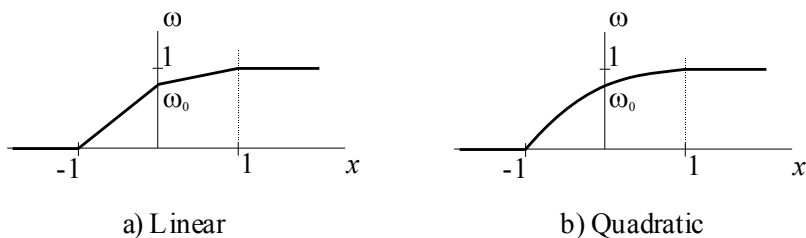


Figure 4.6-6: Gap and pressure-based gap conductance

The auxiliary variable x is used to combine the gap and pressure variables into a single independent variable, as follows:

$$x = -\frac{g}{g_{\max}}, \quad g > 0$$

$$x = \frac{p}{p_{\max}}, \quad p > 0$$

so that negative x corresponds to positive gap and positive x corresponds to positive pressure.

The critical clearance gap g_{\max} , critical contact pressure p_{\max} and the gap conductance ω_0 , at initial contact must be defined.

Time function based gap conductance: The gap conductance ω is zero when the contactor node is out of contact. When the contactor node is in contact, the gap conductance is set to time function $f(t - t_{\text{arrival}})$, where t_{arrival} is the solution time at which contact is most recently established, see Figure 4.6-7.

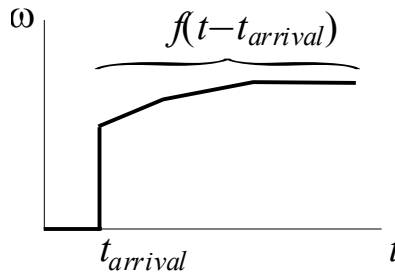


Figure 4.6-7: Time function based gap conductance

Suppose that contact is first established at time t_1 . Then t_{arrival} is set to t_1 , and, as long as contact is established, the gap conductance is set to $f(t - t_1)$. If the contactor node goes out of contact, the gap conductance becomes zero. If the contact node reestablishes contact at time t_2 , then t_{arrival} is set to t_2 , and, as long as contact is

established, the gap conductance is set to $f(t - t_2)$.

The arrival time $t_{arrival}$ is automatically set separately for each contactor node.

The gap conductance time function $f(t - t_{arrival})$ must cover the full solution time.

4.6.1.3 Specification of input

The heat transfer coefficient \hat{h} is specified using parameter HHATTMC of the CGROUP commands. The heat transfer coefficient for a single contact pair can be specified using parameter HHATTMC of the CONTACTPAIR command.

The gap conductance option and associated parameters are defined using the GAP-CONDUCTANCE command.

The default gap conductance for a contact group can be specified using parameter GAPC-NAME of the CGROUP commands, and the gap conductance for a single contact pair can be specified using the parameter GAPC-NAME of the CONTACTPAIR command.

If no gap conductance is specified for a contact group (or a contact pair), then contact-based gap conductance is used.

4.6.2 Heat generation due to friction

4.6.2.1 Theory

Figure 4.6-8 shows heat generation due to friction onto a contact pair.

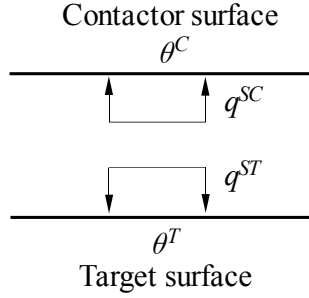


Figure 4.6-8: Heat generation onto a contact pair

The rate of frictional energy dissipation per unit area is

$$\dot{e}_f^S = T_f \dot{u} \quad (4.6-6)$$

in which T_f is the magnitude of the friction traction and \dot{u} is the slip velocity, see Section 4.3.

Some or all of the frictional energy dissipation can be used as heat generation.

The heat flux per unit area applied to the contactor surface is

$$q^{SC} = f_C \dot{e}_f^S \quad (4.6-7)$$

and the heat flux per unit area applied to the target surface is

$$q^{ST} = f_T \dot{e}_f^S \quad (4.6-8)$$

In these equations, f_C and f_T are parameters to account for possible losses. The following relations must hold in order for the modeling to be physically correct (no net energy generated):

$$0 \leq f_C \leq 1, \quad 0 \leq f_T \leq 1, \quad 0 \leq f_C + f_T \leq 1 \quad (4.6-9)$$

Using the same reasoning as that used in Section 4.6.1, the above equations are included in the principle of virtual temperatures as the term

$$\int_{S_C} q^{SC} \delta\theta \, dS + \int_{S_T} q^{ST} \delta\theta \, dS \quad (4.6-10)$$

Since the target contact area is the same as the contactor contact area, this term is rewritten as an integral over just the contactor surface.

$$\int_{S_C} \dot{e}_f^S \left(f_C \delta\theta^C + f_T \delta\theta^T \right) dS \quad (4.6-11)$$

Because the contactor surface is discretized using contactor nodes, the integral is replaced by a sum:

$$\sum_{\text{Contactor nodes}} \dot{e}_f^S A_c \left(f_C \delta\theta^C + f_T \delta\theta^T \right) \quad (4.6-12)$$

where A_c is the contactor node area. The physical situation corresponding to equation (4.6-11) is shown in Figure 4.6-9.

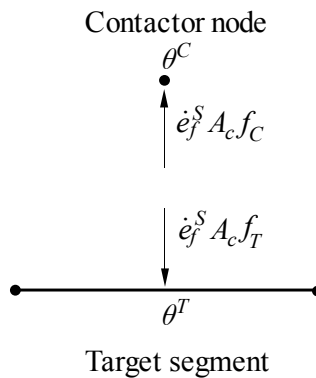


Figure 4.6-9: Heat generation onto a contactor node and target segment

The term $\dot{e}_f^s A_c$ is the rate of frictional energy dissipation at the contactor node. If $f_c = f_r = 0.5$, then half of the frictional energy dissipation becomes heat generated onto the contactor node, and half of the frictional energy dissipation becomes heat generated onto the target segment.

4.6.2.2 Specification of input

Heat generation due to friction is always calculated in thermo-mechanical-coupling analysis.

Coefficients f_c, f_r are specified using parameters FCTMC, FTTMC of the CGROUP commands. These coefficients can also be specified for a single contact pair using parameters FCTMC, FTTMC of the CONTACTPAIR command. The default values of FCTMC, FTTMC are FCTMC=FTTMC=0.5.

In Cat 1 contact, if FCTMC=FTTMC=0 are entered, then FCTMC=FTTMC are reset to 0.5. FCTMC=FTTMC=a small number can be used to specify negligible heat generation due to friction.

In Cat 2 contact, if FCTMC=FTTMC=0 are entered, FCTMC and FTTMC are not reset. FCTMC=FTTMC=0 can be used to specify no heat generation due to friction.

4.7 Contact group features

In this section, we describe the features available within contact groups.

All features pertaining to the rigid-target contact algorithm are described in Section 4.29 and are not described here.

4.7.1 Contact group birth/death

The contact group birth feature activates a contact group at a specific time, while the contact group death feature disables a contact group at a specific time. A 0.0 birth time means that the

contact group starts active at the beginning of the analysis, and a death time less than or equal to the birth time means that the contact group does not die.

The contact group birth and death times are set via the TBIRTH and TDEATH parameters of the CGROUP commands.

The contact group birth and death times also provide defaults for the contact pair birth and death times.

4.8 Contact surfaces

The contact surfaces should be defined as regions that are initially in contact or that are anticipated to come into contact during the solution.

4.8.1 Contact surfaces for segment contact

4.8.1.1 2-D contact surfaces

2-D contact surfaces lie in the same plane as 2-D element groups. This plane is controlled by the 2DPL-AX parameter of the MASTER command.

Typical two-dimensional contact surfaces are shown in Fig. 4.8-1. Each contact surface consists of contact segments formed by connecting successive nodes in the contact surface.

2-D contact surfaces are either axisymmetric, plane strain or plane stress.

Axisymmetric: In both Cat 1 and Cat 2 contact, axisymmetric contact surfaces represent one radian of the domain (same assumption as is used for 2-D axisymmetric solid elements).

Plane strain: In both Cat 1 and Cat2 contact, the thickness of 2D plane strain contact surfaces is always assumed to be 1 (same assumption as is used for 2-D solid plane strain elements).

Plane stress: In Cat 1 contact, the thickness of 2D plane stress contact surfaces is always assumed to be 1 (unit thickness). In Cat 2 contact, the thickness of 2D plane stress contact surfaces can be entered using the THICKNESS parameter of the CGROUP CONTACT2 command (default is 1).

Closed and open surfaces: A 2-D contact surface is a closed surface if all the contact segments on the surface taken together form a closed path (see Fig. 4.8-1, contact surface 3). If the segments do not form a closed path, then the contact surface is open (see Fig. 4.8-1, contact surfaces 1 and 2).

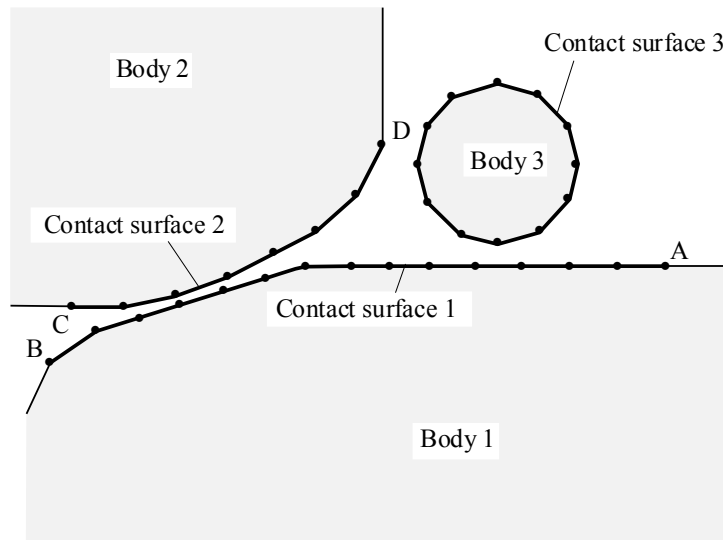


Figure 4.8-1: Typical 2-D contact surfaces

4.8.1.2 3-D contact surfaces

A 3-D contact surface is made up of an assemblage of 3-D contact segments (faces) either on solid elements, shell elements, plate elements, or attached to rigid nodes. See Fig. 4.8-2 for an illustration. The segments of a 3-D contact surface can be connected to each other, disconnected from each other, or some combination of connected and disconnected.

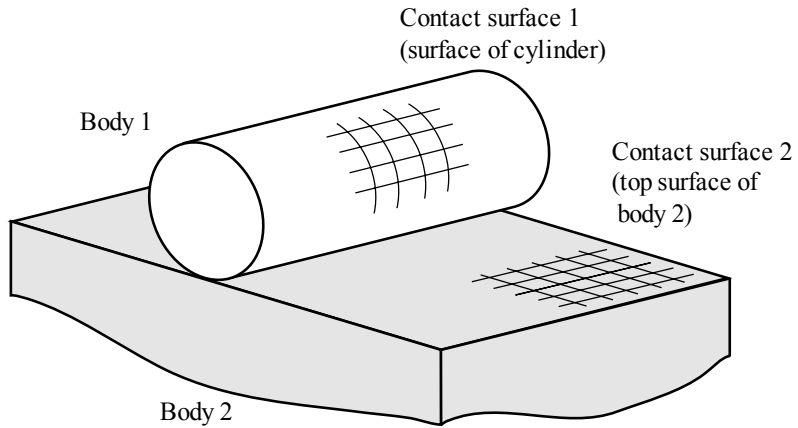
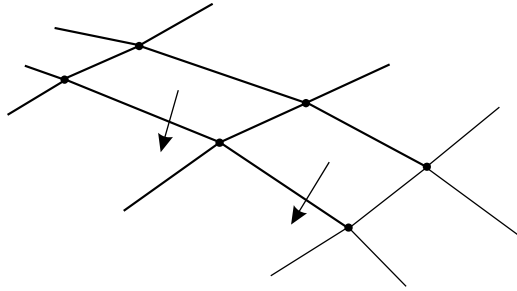
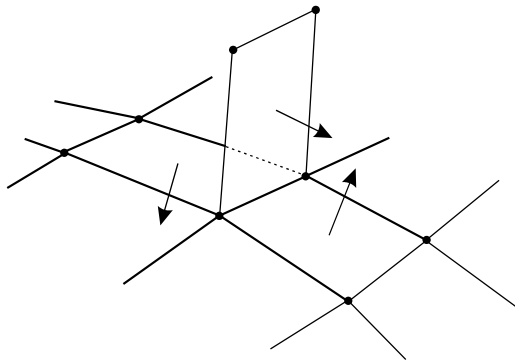


Figure 4.8-2: Typical 3-D contact surfaces

When the surface is used as a target surface, it is recommended that each edge of the surface be shared by at most two segments, see Fig. 4.8-3.



a) Each edge is shared by at most two target segments, good



b) An edge is shared by three or more target segments, bad

Figure 4.8-3: Target surface edges

4.8.1.3 Rigid and flexible surfaces

Rigid surfaces have no underlying elements and therefore no flexibility apart from rigid body motions. All of the nodes of a rigid surface must be either fixed, have prescribed displacements, or be rigidly linked to a master node.

4.8.1.4 Contact segment normals

Each contact segment within a contact surface has a normal direction \mathbf{n} that points into the overlap direction, as shown in Figs. 4.8-4 and 4.8-5.

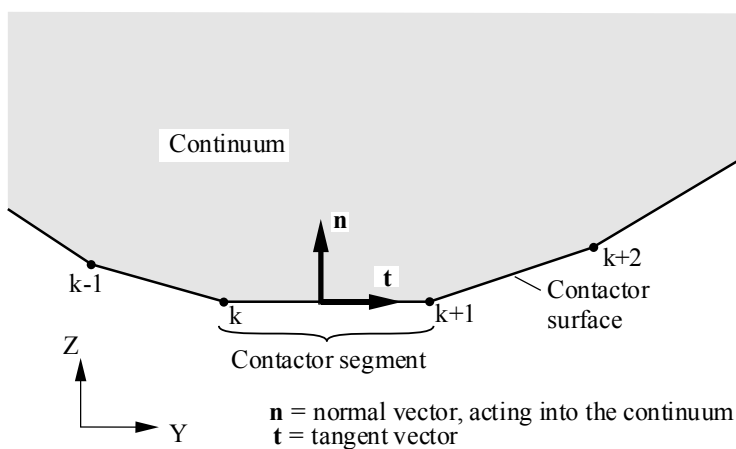


Figure 4.8-4: Normal vector of a 2-D contact segment

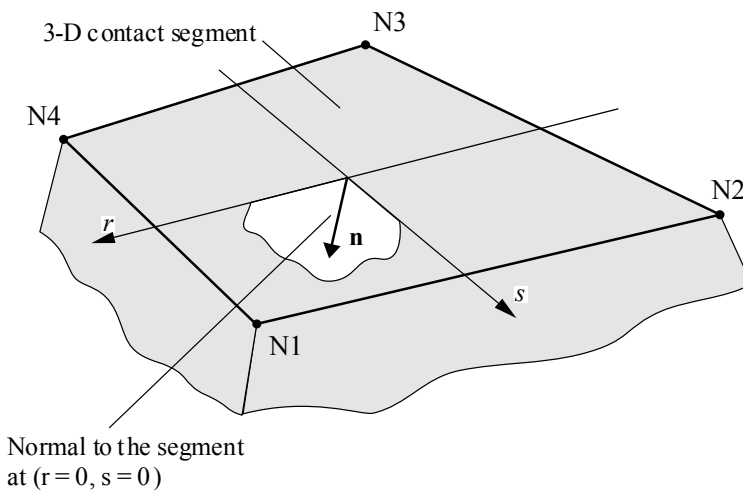
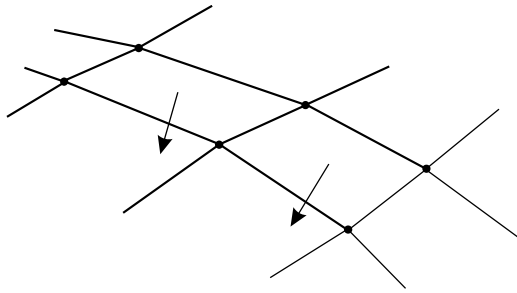
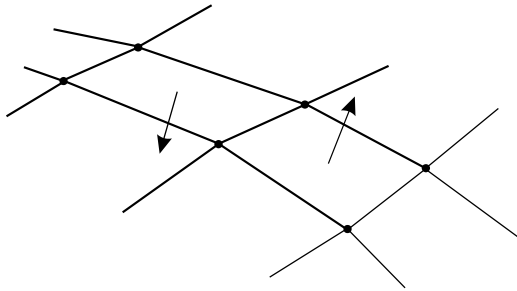


Figure 4.8-5: Normal vector of a 3-D contact segment

For 3D surfaces, it is recommended that the normal vectors of adjacent segments not be flipped, see Fig 4.8-6. Normal vectors of adjacent segments should, as far as possible, point in nearly the same directions.



a) Segment normals are not flipped, good



b) Segment normals are flipped, bad

Figure 4.8-6: Normal vectors of adjacent 3-D contact segments

4.8.1.5 Continuous and segment normals

There are two options regarding the normal used at an isoparametric coordinate within a target segment:

Continuous normals: The program calculates nodal normals as averages of all of the normals from the attached segments. Then the nodal normals are interpolated within the target segment. In this way, the normal direction is continuous across neighboring segments.

Segment normals: The segment normal is used for all isoparametric coordinates within the target segment. The normal direction need not be continuous across neighboring segments.

The discontinuity between the normals of the neighboring segments sometimes leads to convergence difficulties.

Fig 4.8-7 illustrates these options.

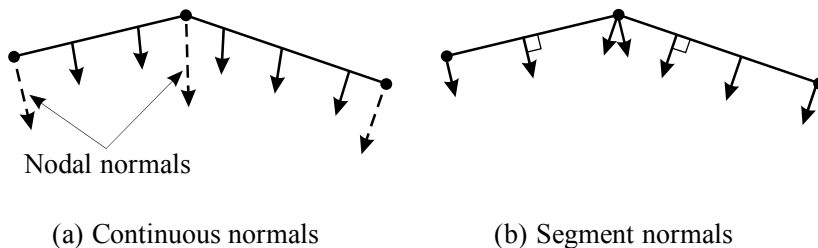


Figure 4.8-7: Contact surface normals

The continuous normals feature is specified using the `CONTINUOUS-NORMAL` parameter of the `CGROUP` commands. Continuous normals is the default setting.

In Cat 1 contact, continuous normals are not available in explicit analysis, and segment normals are always used.

Generally speaking, if the contact surface has no sharp corners, continuous normals should be used. In modeling target surfaces with sharp corners, either use segment normal vectors, or use small segments near the corners, to ensure that the normal vectors for segments near the corners are computed correctly. See Section 4.26.3 for modeling tips related to this feature.

In Cat 1 contact, there is only one option for the computation of the continuous normal:

Average: The continuous normal is the average of the attached segment normals.

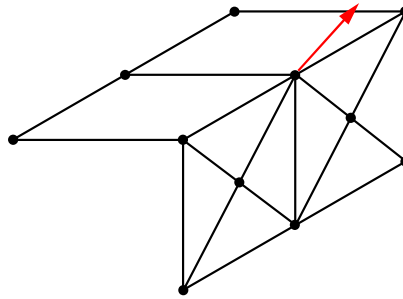
In Cat 2 contact, there is an additional option:

Median: The continuous normal is the median of the attached segment normals. The median normal is midway between the most extreme segment normals.

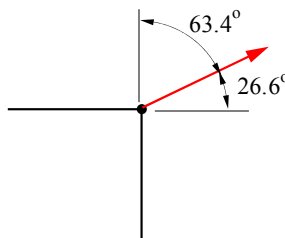
Median continuous normals are selected using $\text{CNCALC}=\text{MEDIAN}$ in the CONTACT-CONTROL or CGROUP commands.

For 2D contact surfaces, the median and average normals are the same. Therefore the median continuous normals feature is useful only for 3D contact surfaces.

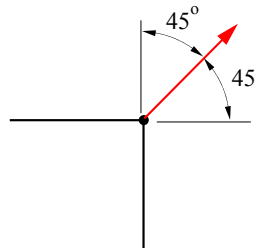
The median continuous normal is most useful when the attached segment normals point in greatly different directions, since then the median continuous normal has a smaller maximum angular deviation from the attached segment normals than does the average continuous normal. Fig 4.8-8 shows an example in which the average and median continuous normals are different.



(a) Contact surface segments, parallel projection



(b) Average continuous normal,
side view



(c) Median continuous normal,
side view

Figure 4.8-8: Average and median continuous normals

The default in Cat 2 contact is to use average continuous normals.

4.8.1.6 Single-sided contact

For single-sided contact (see Fig. 4.8-9), one side of the contact surface is assumed to be internal and the other side to be external. Any contactor node within the internal side of a target surface is assumed to be penetrating. This single-sided option is ideal for contact surfaces on the faces of solid elements since in that case it is clear that one side is internal to the solid while the other is external. In this case, the external side can usually be predicted from the geometry. This option is also useful for shells when it is known that contact will definitely occur from one direction. In this case, however, the program cannot intuitively predict the internal side of the contact surface.

Single-sided contact is selected using CGROUP ...
PENETRATION-ALGORITHM=ONE (the default).

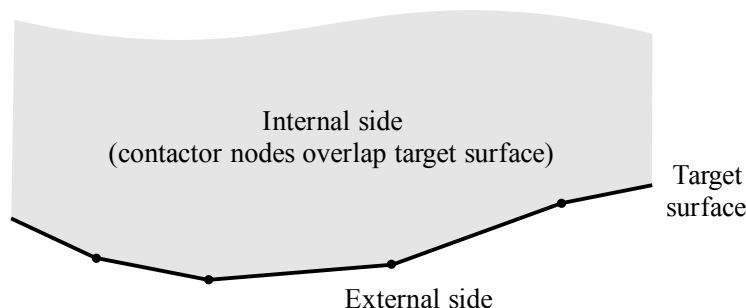


Figure 4.8-9: Single-sided contact surface

4.8.1.7 Double-sided contact

In double-sided contact (see Fig. 4.8-10), there are no internal or external sides. The contactor surface nodes in this case are prevented from crossing from one side of the target contact surface to the other during solution. This option is more commonly used for shell-based contact surfaces. If a contactor node is on one side of the target surface at time t , it will remain on the same side at time $t + \Delta t$.

Double-sided contact is selected using CGROUP ... PENETRATION-ALGORITHM=TWO. Double-sided contact is only available for 3D contact.

If double-sided contact is selected for a contact group in Cat 2 contact, the same contact algorithms are used as in Cat 1.

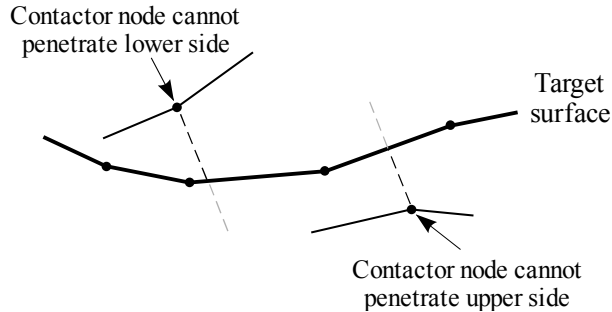


Figure 4.8-10: Double-sided contact surface

4.8.1.8 Contact surface extension

A contact surface used as a target surface can be enlarged beyond its geometric bounds, so that contactor nodes that slip outside the target can still be considered in contact. This feature is useful where the edge of the contactor and target surfaces coincide, as shown in Fig. 4.8-11. Each target segment is enlarged by an amount equal to the contact surface extension factor multiplied by the length of the segment.

The contact surface extension factor is specified using the CS-EXTENSION parameter of the CGROUP commands.

In Cat 1 contact, all target segments are enlarged by the CS-EXTENSION factor. In Cat 2 contact, only the target segments with free edges (that is, edges attached to only one target segment) are enlarged by the CS-EXTENSION factor, and the enlargement is in the direction of the free edges.

The default for CS-EXTENSION is 0.001. If a value for CS-EXTENSION less than 0.001 is entered, a value of 0.001 is used.

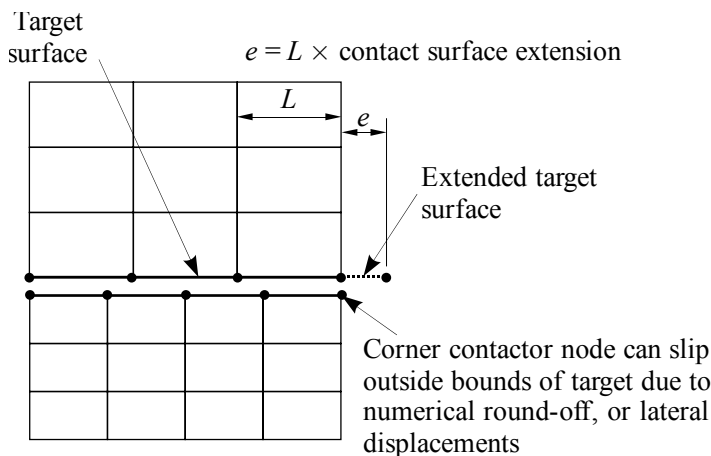


Figure 4.8-11: Contact requiring contact surface extension

4.8.1.9 Contact segments

In 2D contact, contact segments are either linear (2-node) or quadratic (3-node). These segment types correspond to the edges of linear (3- and 4- node) 2D elements and quadratic (up to 9 nodes) 2D elements.

In 3D contact, contact segments are either linear (3 or 4 nodes) or quadratic (up to 9 nodes). These segment types correspond to the faces of linear (4 to 8 node) 3D elements and quadratic (up to 27 nodes) 3D elements.

In Cat 1 contact, the above description applies when the “new” contact surface representation is used. In Cat 1 contact, there is also the option of using the “old” contact surface representation. In the “old” contact surface representation, only the linear contact segments are supported. In addition, the traction calculation is somewhat different and the output is segment-based. The contact surface representation is specified using the CSTYPE parameter of the CONTACT-CONTROL command. The new contact surface representation is the default.

4.8.1.10 Contact bodies

A contact body is a special type of contact surface, in which the nodes of the contact surface are not connected by segments (see Fig. 4.8-12). The contact body can only be the contactor in a contact pair, and the target has to be a contact surface (i.e. the target cannot be a contact body).

Contact bodies are useful for modeling metal cutting applications where the internal nodes of a body can also come in contact once the outer ones are removed. Contact bodies can also be used to model 3D contact problems involving beams (since the beams cannot define a 3D contact surface).

Contact bodies are defined using the `CONTACTBODY` command.

Contact bodies cannot be used in tied contact.

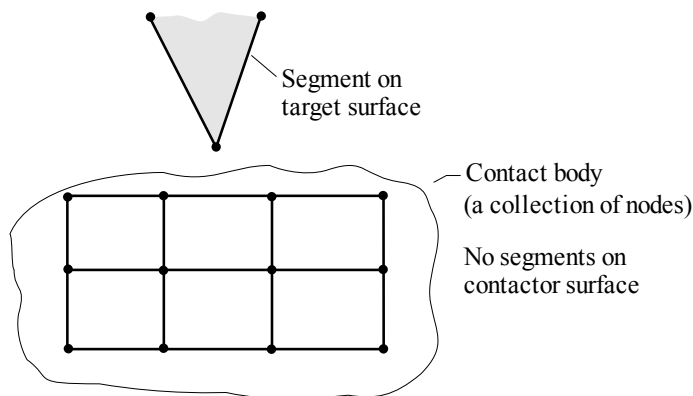


Figure 4.8-12: Contact body

4.8.1.11 Contact surface offsets

Penetration of a contact surface occurs when the target segment is penetrated. However, an offset distance can be specified which causes the target segments to be offset from the plane defined by the contact surface nodes. In the case of double-sided contact, the offset creates two separate surfaces above and below the reference surface. Fig. 4.8-13 shows the possibilities for single and double-sided contact. Note that the offset distance should be small

compared to the contact surface length.

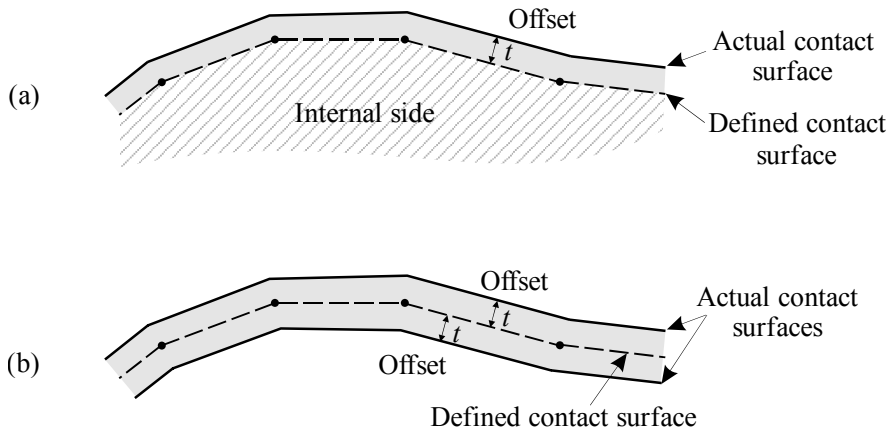
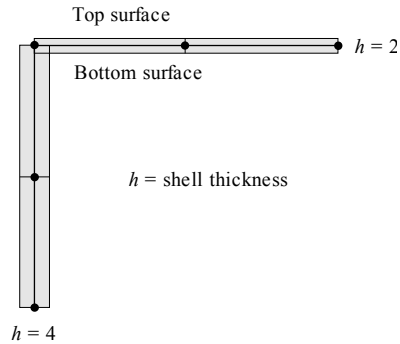


Figure 4.8-13: Contact surface offsets for: (a) single-sided contact, and (b) double-sided contact

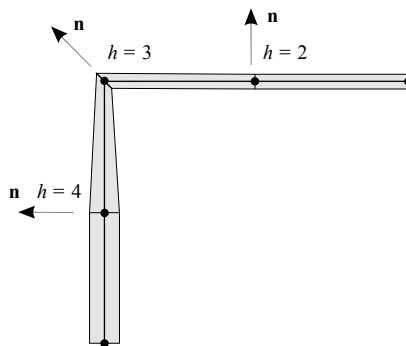
Offsets for a whole contact group are specified in the CGROUP commands, using parameters OFFSET-TYPE and OFFSET. Offsets for a specific contact surface can also be defined using the CS-OFFSET command. If one of the contact surfaces has a defined offset, it will overwrite the contact group offset.

The option OFFSET-TYPE=TRUE can be used to set half of the true shell thickness as the offset distance. When this option is used, the shell thickness at a contact node is computed as the average of all of the shell thicknesses at that node. Fig 4.8-14 shows an example. Notice that at the shell-shell intersection, when continuous normals are used, the average thickness of 3 is measured in the direction of the continuous normal.

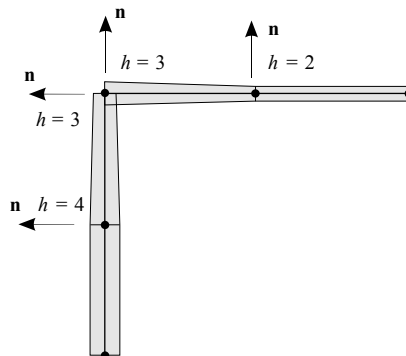
Fig 4.8-15 shows an example in which only some of the shell elements are attached to contact segments. Notice that all of the shell elements, including the shell elements not attached to contact segments, are used in the calculation of the average thickness.



(a) Shell structure, as seen from the side



(b) Contact surface offsets, continuous normals



(c) Contact surface offsets, segment normals

Figure 4.8-14: Contact surface offsets used for shell structure

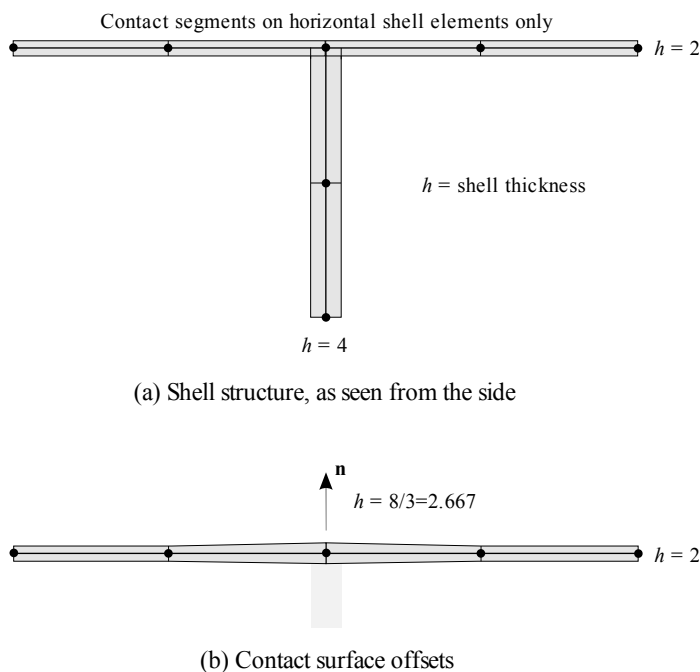


Figure 4.8-15: Contact surface offsets used for shell structure, not all shell elements are attached to contact segments

The use of contact surface offsets in double-sided contact is not recommended.

4.8.1.12 Contact surface depth

By default, the overlap region of a target surface extends for an infinite distance below the contact surface (for single-sided contact). However, a contact surface depth can be defined (by setting the DEPTH parameter in the CGROUP commands), below which the contact surface is no longer active. The default DEPTH=0.0 results in an infinite contact depth extension. Fig. 4.8-16 shows some of the possibilities.

Contact surface is used in self-contact.

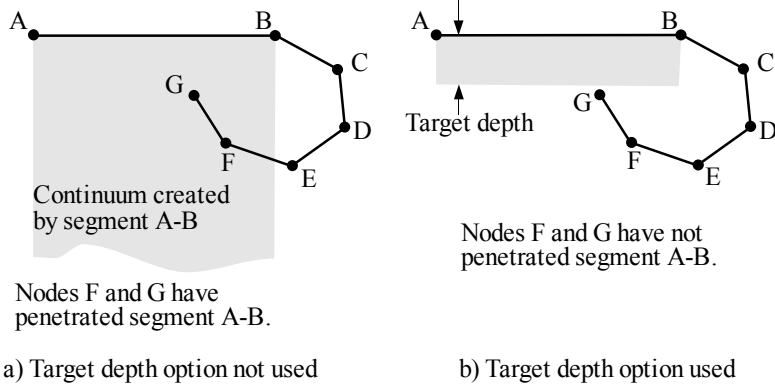


Figure 4.8-16: Contact surface depth

For Cat 2 contact, self-contact surfaces automatically create a depth value if DEPTH is not user-specified.

4.8.1.13 Contact slip loads

The contact slip load adds a prescribed velocity to the contact surface. Conceptually, the contact surface is assumed to be undergoing a prescribed rotation with a given axis of rotation, an angular velocity and a time function. The velocity corresponding to the prescribed rotation is termed the contact slip loading velocity.

For more information about contact slip loads, see Section 5.11.

In Cat 1 contact, contact slip loads cannot be used in explicit analysis.

4.8.1.14 Automatic creation of contact surfaces

Contact surfaces and contact pairs can be automatically created between two bodies, using the CONTACT3-SEARCH command. All surfaces within a user-specified distance of each other are accounted for.

4.8.2 Contact surfaces for node to node contact

When node-node contact is used, each node of the contactor surface is paired with a node on the target contact surface (a node-node contact pair). A node on the contactor surface can come into contact with the infinite plane passing through the paired node on the target surface; the infinite plane is perpendicular to the normal direction indicating the direction within the target body (Fig. 4.8-17). This normal direction, and hence the infinite plane, is constant during the response. The normal directions can be based on the geometry of the target segments or on the vector connecting each contactor and target nodes.

Node-node contact is selected using CGROUP ...
NODETONODE=YES.

If node-node contact is selected for a contact group in Cat 2 contact, the same contact algorithms are used as in Cat 1.

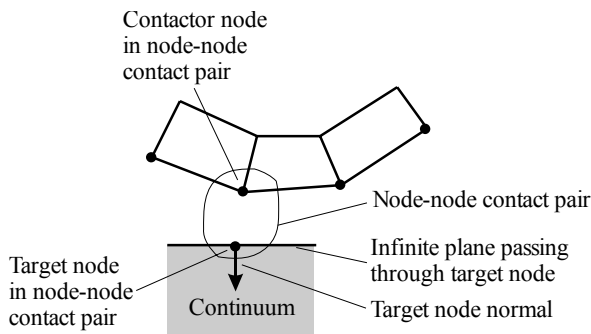


Figure 4.8-17: Node-to-node contact

4.8.2.1 Analytical rigid targets for node-node contact

There is the possibility of describing target contact surfaces as smooth analytically described surfaces for some simple types of geometry. These analytically described surfaces are termed “analytical rigid targets” (ARTs).

The smooth surface representation with ARTs allows for more accurate and more easily converging solutions, especially when friction is present. Most of the applications of ARTs are in metal forming problems.

The possible geometries are infinite planes, spheres, circles, infinite cylinders or strips (the intersections of infinite cylinders with the YZ plane for 2-D problems).

Each ART is connected to one reference node, which defines the motion of the ART. ARTs can move translationally and rotate with the reference node as a rigid body, but cannot deform. Usually the motion of the reference node is controlled with prescribed displacements. In general the reference node can be any node in the model, free to move or with boundary conditions or constraints.

There can be several ARTs in the model, but only one ART in each contact group. The usual node-to-node contact can be used with an ART in a contact group. Contact can either be frictionless or with friction, and all types of static and dynamic analyses in ADINA are possible when ARTs are present.

There are the following limitations. The ARTs are not plotted by the AUI and are therefore invisible. Contact tractions cannot be computed with ART contact.

Analytical rigid targets are defined using the ANALYTICAL-RIGID-TARGET command.

4.9 Contact pairs

A contact pair consists of two contact surfaces that can come into contact during the solution. One of the contact surfaces in the pair is selected to be the contactor surface and the other contact surface to be the target surface. In the case of self-contact, the same surface is selected to be both contactor and target in the contact pair. (Self-contact is when a contact surface is expected to come into contact with itself during the solution.)

Each node on the contactor surface (contactor node) is associated with a target segment on the target surface, if possible. Contact

conditions are determined for each contactor node. For example, if a no-overlap normal contact response is selected, then the contactor nodes are prevented from penetrating the target segments.

In a contact group in which a contactor surface is used in more than one contact pair, a contactor node can only be in contact with one of the target surfaces of the contact pairs.

The interaction of contactor nodes and target segments is described in much more detail in Sections 4.10 to 4.22.

Fig. 4.9-1 shows the effect of contactor surface and target surface selection on the different contact configurations.

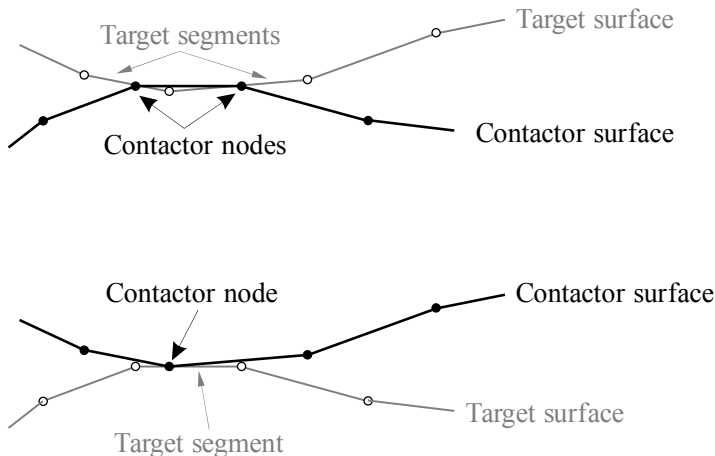


Figure 4.9-1: Contactor and target selection

At least one of the two contact surfaces in a contact pair must not be rigid. If one surface is rigid, this surface should, in most cases, be the target surface.

In explicit dynamic analysis, both contactor and target surfaces can be rigid if the penalty algorithm is used. Otherwise, the same restriction mentioned above applies.

4.9.1 Normal contact properties

In Cat 2 contact, the normal contact response properties can be assigned to individual contact pairs. This assignment is done by defining the normal contact response properties in a CNORMAL command, then specifying the CNORMAL value in the contact pair definition:

```
CNORMAL BASIC 1 ...  
CONTACTPAIR 1 ... CNORMAL=1
```

4.9.2 Frictional contact properties

For frictional contact with the Coulomb friction law the coefficient of friction can be assigned to individual contact pairs.

```
CONTACTPAIR 1 ... FRICTION=...
```

In Cat 1 contact, the parameters of friction models 1 and 2 can be assigned to individual contact pairs using the COULOMB-FRICTION command.

In Cat 2 contact, a friction model can be assigned to an individual contact pair using one of the CFRICTION commands:

```
CFRICTION COULOMB 1 ...  
CFRICTION LAW1 2 ...  
CONTACTPAIR 1 ... CFRICTION=1  
CONTACTPAIR 2 ... CFRICTION=2
```

4.9.3 Relative contact damping properties

In Cat 2 contact, the relative contact damping properties can be assigned to individual contact pairs. For example

```
CRELDAMP BASIC 1 ...  
CRELDAMP BASIC 2 ...  
CONTACTPAIR 1 ... CRELDAMP=1  
CONTACTPAIR 2 ... CRELDAMP=2
```

4.9.4 Multiphysics properties

The multiphysics properties associated with contact heat transfer can be assigned to individual contact pairs.

4.9.5 Self-contact

It is allowed to use the same contact surface for the contactor surface and target surface in the same contact pair.

In Cat 1 contact, a contactor node cannot be in contact if the contactor node belongs to a target segment that is in contact.

In Cat 2 contact, a contactor node can be in contact if the contactor node belongs to a target segment that is in contact. Therefore a non-zero contact surface compliance should be used in conjunction with self-contact.

4.9.6 Symmetric contact

Symmetric contact pairs can be defined, where in one contact pair surface A can be the contactor and surface B the target, and in another contact pair surface B is the contactor and surface A is the target.

A non-zero contact surface compliance should always be used with symmetric contact pairs. Without compliance, the system of contact equations can become overconstrained, that is, the same contact equation can be present more than once in the system of contact equations, see Section 4.19.1.4 for a description of overconstraining. When this happens, the global system of equations becomes singular and the solution may diverge.

4.9.7 Contact pair birth/death

The contact pair can be assigned a birth and/or death time. For solution times prior to the birth time, or after the death time, the contact pair is ignored during the contact calculations.

By default, the birth and death times of a contact pair are the birth and death times of the contact group.

4.10 Contact interactions and contactor elements

In Sections 4.11 to 4.22, we describe all features involving contact interactions between contact surfaces.

Generally, interactions between contact surfaces are described in terms of contactor elements.

A contactor element is the combination of a contactor node and target segment, as shown in Fig. 4.10-1.

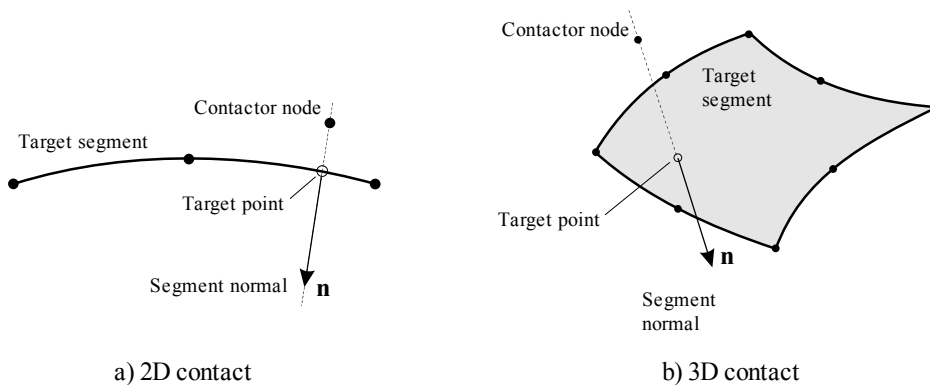


Figure 4.10-1: Contactor elements in 2D and 3D contact

4.11 Large displacement contact

Large displacement contact is selected by setting `DISPLACEMENT=LARGE` in the `CGROUP` commands. Large displacement contact is the default.

In large displacement contact, the contactor elements are determined in each equilibrium iteration, and the contactor nodes can undergo any amount of sliding.

The contactor elements are determined using search algorithms.

4.11.1 Contact search for Cat 1 contact

During each equilibrium iteration, the most current geometry of the contactor and target surfaces is used to determine the contactor elements.

For single-sided contact, the determination of the contactor element for contactor node k consists of a contact search. The contact search starts by identifying all possible target surfaces where node k can come into contact. For each of these target surfaces:

- Find the closest target node n to node k .
- Find all the target segments attached to node n .
- Determine if node k is in contact with any of these segments.
- If node k is in contact, update the information.
- If no appropriate target segment is detected, the contact search is expanded beyond the target segments attached to node n .

The Cat 1 contact search is shown in Fig. 4.11-1.

For 3-D single-sided contact in explicit analysis only, a fast search algorithm is available (Command CONTACT-CONTROL 3D-SEARCH=FAST). This algorithm is significantly faster when the contact pair has a very large number of contactor nodes and target segments.

For double-sided contact, the contact search algorithm uses time tracking and checks whether the contactor node penetrated a target segment between times t and $t + \Delta t$.

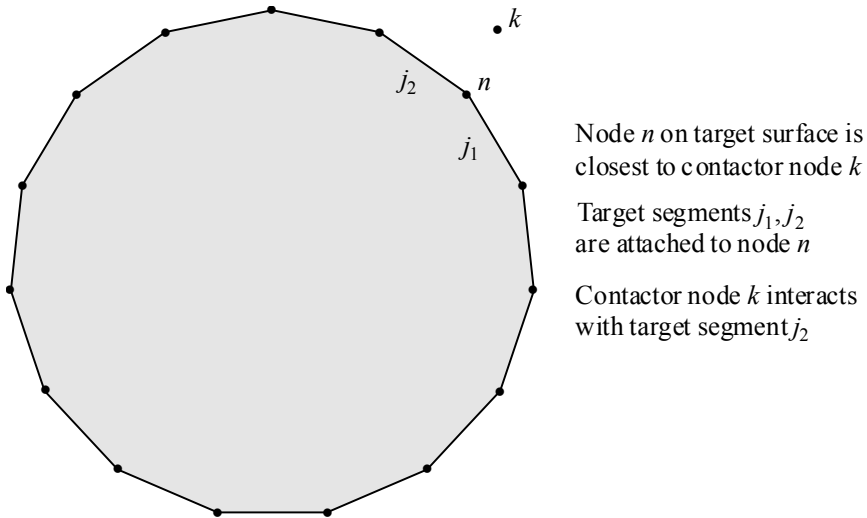


Figure 4.11-1: Contact search for Cat 1 contact

When no-overlap contact is used, the Cat 1 contact search is modified as follows. If contact node k is in contact in a contact group, then contact node k is not permitted to be in contact in any other contact group. In self-contact, contact node k is not permitted to be in contact with a target segment that contains contact node k . These modifications are made to avoid overconstraining, see Section 4.19.1.4 for a description of overconstraining.

The Cat 1 contact search can be incorrect for certain target surface meshings, as shown in Fig 4.11-2. Contactor node k should interact with target segment j_3 . However the closest node to k is not attached to j_3 , instead the closest node to k is attached to j_1 and j_2 . The contactor node interacts with target segment j_1 , which is incorrect.

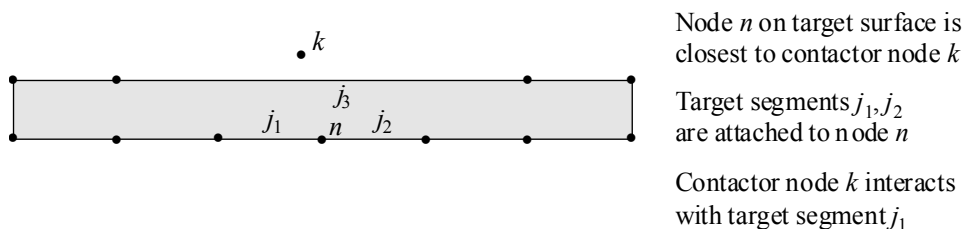


Figure 4.11-2: Example of incorrect contact search for Cat 1 contact

4.11.2 Contact search for Cat 2 contact

During each equilibrium iteration, the most current geometry of the contactor and target surfaces is used to determine the contactor elements.

For single-sided contact, the determination of the contactor element for contactor node k consists of a contact search. The contact search starts by identifying all target segments j in which contactor node k is in the field of view of j . The closest target segment to contactor node k is chosen. This process is shown in Fig. 4.11-3.

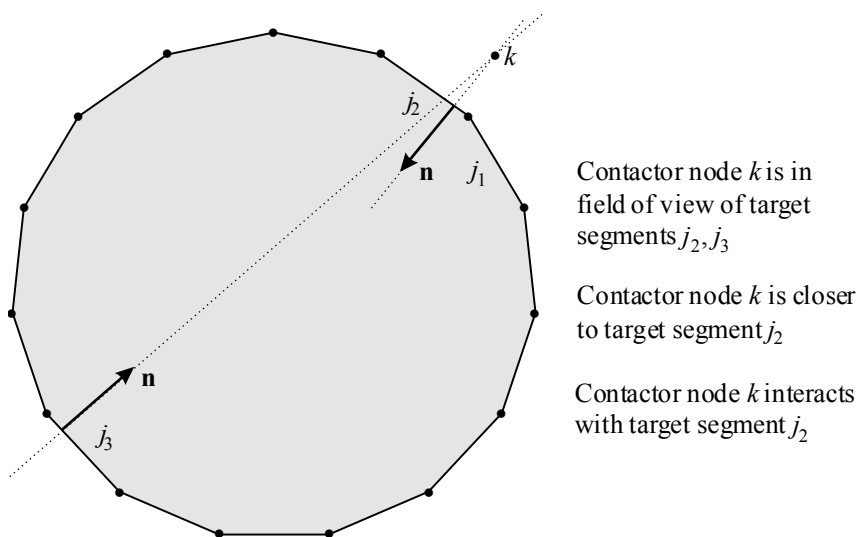


Figure 4.11-3: Contact search for Cat 2 contact

When no-overlap contact is used, the Cat 2 contact search is not modified to avoid overconstraining. If overconstraining is possible (see Section 4.19.1.4 for a description of overconstraining), for example due to a contactor node being in contact in multiple contact groups, compliant contact should be used.

Cat 2 contact search is SMP parallelized. Therefore increasing the number of threads in the run reduces the wall-clock time required for contact search.

The Cat 2 contact search is more robust than the Cat 1 contact search. Fig. 4.11-4 shows the same example as Fig. 4.11-2. The Cat 2 contact search determines the correct target segment for contactor node k .

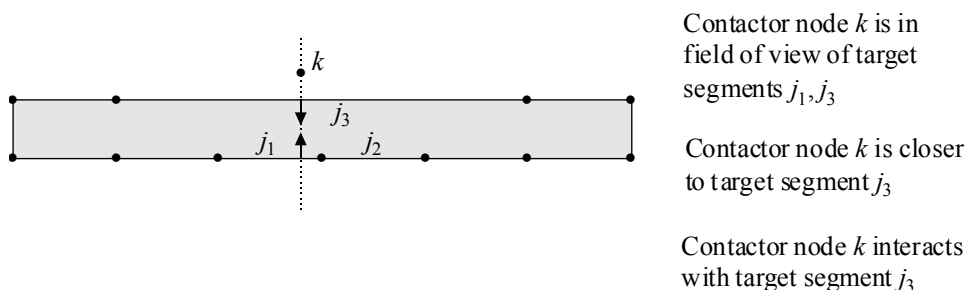


Figure 4.11-4: Example Fig. 4.11-2 works correctly with Cat 2 contact

The Cat 2 contact search is significantly more expensive than the Cat 1 contact search, because all of the target segments for which contactor node k is in the field of view need to be determined, see Fig. 4.11-5.

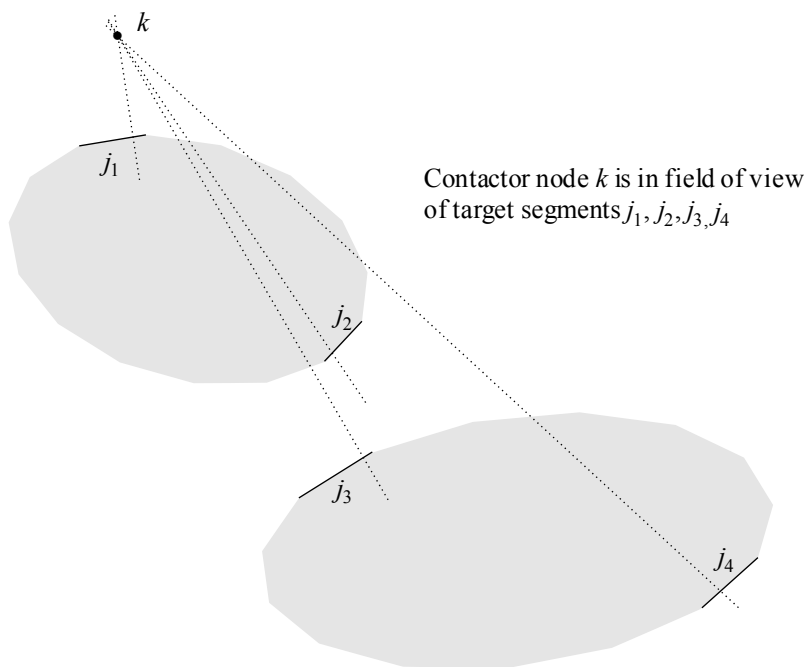


Figure 4.11-5: Multiple target surfaces in Cat 2 contact

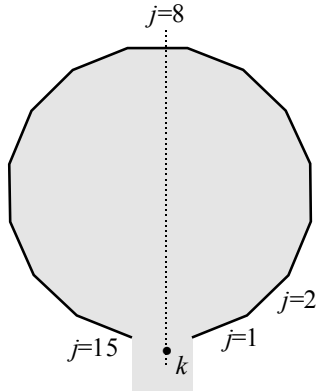
Cat 2 contact search and open target surfaces

The Cat 2 contact search can find incorrect target segments when the target surfaces are open. The following figures show examples of incorrect target segments in Cat 2 contact.

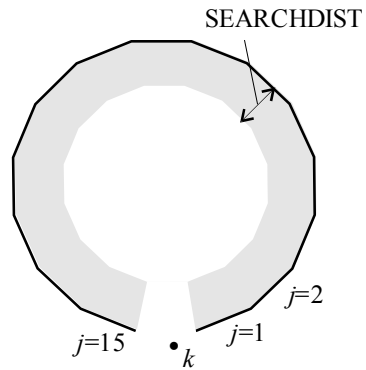
In Fig 4.11-6, contactor node k is in the field of view of target segment 8. Therefore contactor node k interacts with target segment 8, incorrectly. Note that if target segment 16 had been defined, the target surface would have been closed and contactor node k would have interacted with target segment 16, with a gap. The search would have been correct in this case.

The SEARCHDIST feature described below can also prevent the incorrect search.

Contactor node k is in field of view
of target segment 8



a) SEARCHDIST feature not used



b) SEARCHDIST feature used

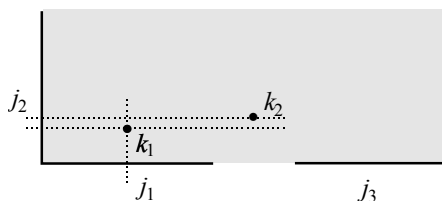
Figure 4.11-6: Incorrect contact search in Cat 2 contact

In Fig 4.11-7, contactor node k_1 is in the field of view of target segments j_1, j_2 , which is correct. Because j_1 is closest to k_1 , k_1 interacts with j_1 . However contactor node k_2 is in the field of view of target segment j_2 , and there is no closer target segment for which k_2 is in the field of view. So contactor node k_2 interacts with j_2 , which is incorrect.

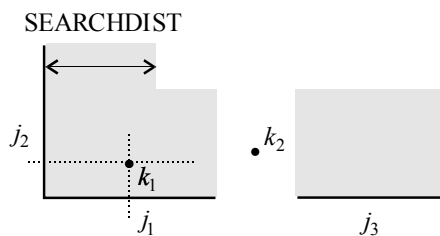
The SEARCHDIST feature described below can prevent the incorrect search.

Contact node k_1 is in field of view
of target segments j_1, j_2 , OK

Contact node k_2 is in field of view
of target segment j_2 , incorrect



a) SEARCHDIST feature not used



b) SEARCHDIST feature used

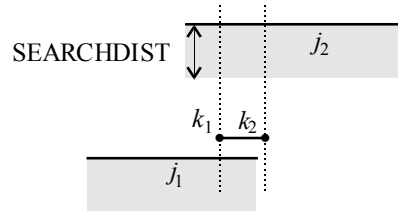
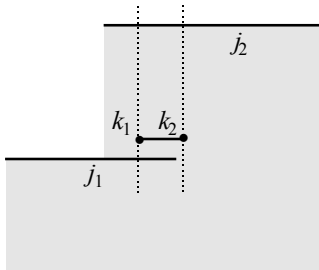
Figure 4.11-7: Incorrect contact search in Cat 2 contact

Fig 4.11-8 shows two disjoint target segments in the same target surface. Contact node k_1 is in the field of view of target segments j_1, j_2 , which is correct. Because j_1 is closest to k_1 , k_1 interacts with j_1 . However contact node k_2 is in the field of view of target segment j_2 , and there is no closer target segment for which k_2 is in the field of view. So contact node k_2 interacts with j_2 , which is incorrect. This situation can occur when the contactor surface containing k_1, k_2 is larger than the target surface containing j_1 . If the target surface containing j_1 were slightly larger, then k_2 would have interacted with target segment j_1 , correctly.

The SEARCHDIST feature described below can prevent the incorrect search.

Contactor node k_1 is in field of view
of target segments j_1, j_2 , OK

Contactor node k_2 is in field of view
of target segment j_2 , incorrect



a) SEARCHDIST feature not used

b) SEARCHDIST feature used

Figure 4.11-8: Incorrect contact search in Cat 2 contact

A more practical example is shown in Fig. 4.11-9. In this example, the rim of the wheel is modeled with target segments. A contactor node near the upper target is not in the field of view of the upper target, but is in the field of view of the lower target. So the contactor node interacts with the lower target. Fig. 4.11-9(c) shows how the SEARCHDIST feature can be used to prevent this unintended interaction.

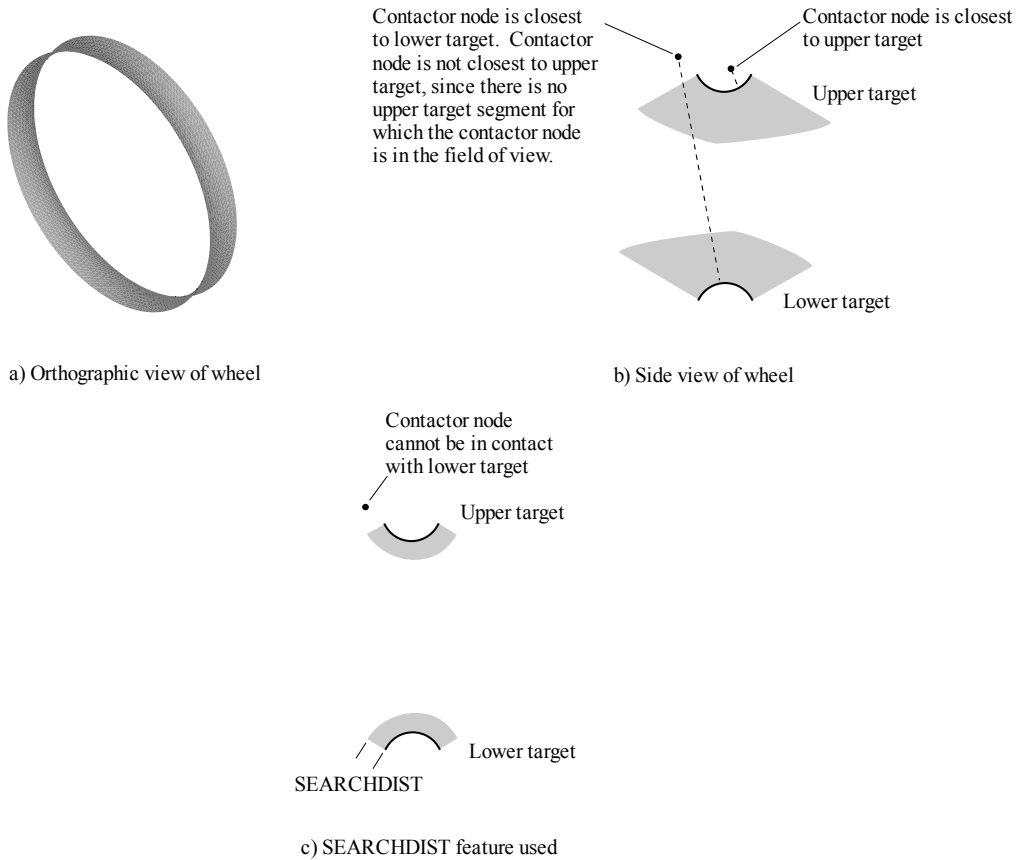


Figure 4.11-9: Incorrect contact search in Cat 2 contact, wheel model

A similar example is shown in Fig. 4.11-10. In Fig 4.11-10(a), there is a gap between the contactor node and the closest target segment, as expected. In Fig. 4.11-10(b), the punch has moved upwards relative to the contactor node. Now there is a large overlap between the contactor node and the closest target segment for which the contactor node is in the field of view. Fig. 4.11-10(c) shows how the SEARCHDIST feature can be used to prevent this unintended interaction, and Fig 4.11-10(d) shows how adding more target segments prevents this unintended interaction.

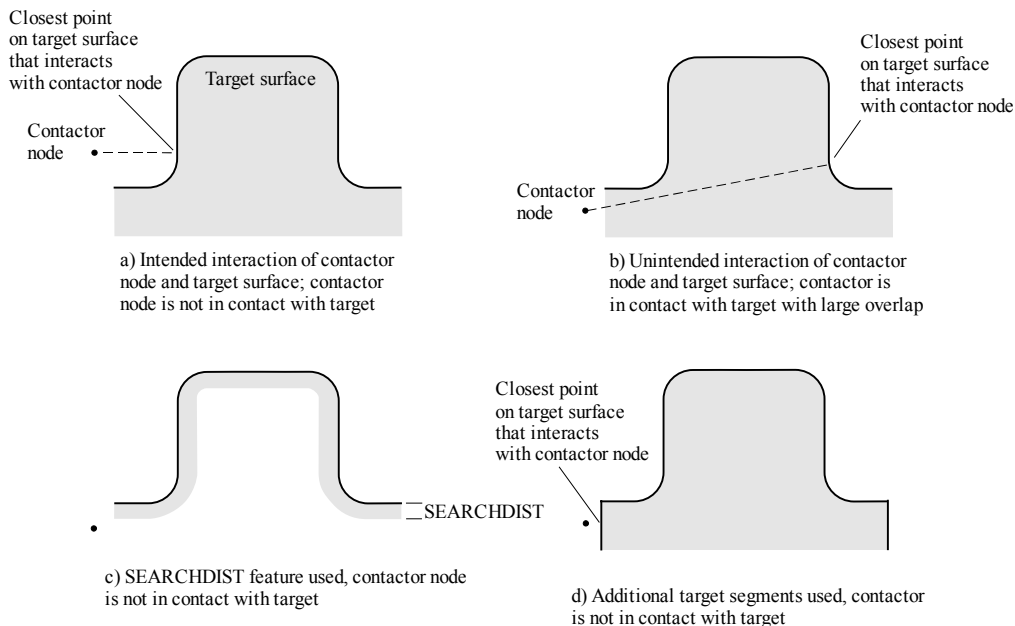


Figure 4.11-10: Incorrect contact search in Cat 2 contact, punch model

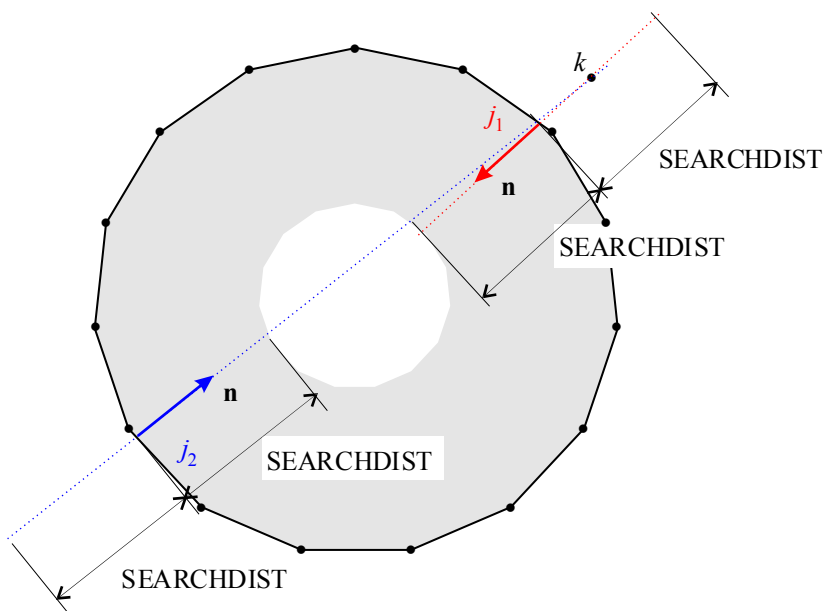
SEARCHDIST parameter

The Cat 2 contact search can be sped up, and incorrect Cat 2 contact searches can be eliminated, using the SEARCHDIST parameter of the CONTACT-CONTROL and CGROUP commands. The SEARCHDIST concept is shown in Fig 4.11-11.

The field of view of a target segment is truncated at distance SEARCHDIST. Therefore a target segment that is further away than SEARCHDIST from the contactor node will not interact with the contactor node.

In many cases, an appropriate value of SEARCHDIST can be chosen from physical considerations.

The incorrect contact searches of Figs. 4.11-6 to 4.11-10 can be eliminated by choosing SEARCHDIST appropriately. The DEPTH feature of Section 4.8.1.12 could also be used, but using SEARCHDIST is much more efficient.



Contactor node k is in field of view of target segment j_1

Contactor node k is not in field of view of target segment j_2

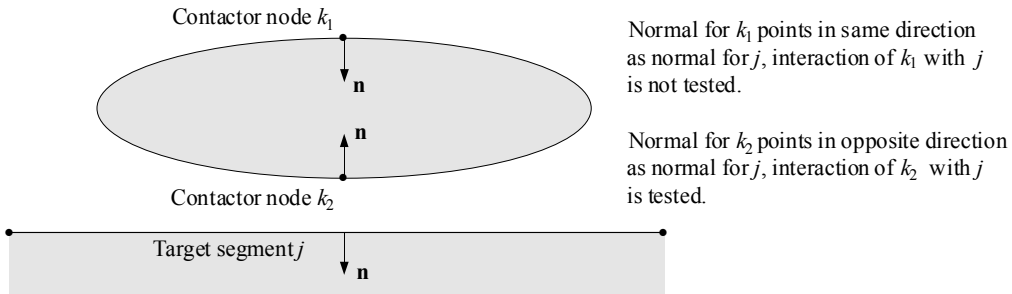
Figure 4.11-11: SEARCHDIST parameter in Cat 2 contact

By default SEARCHDIST=0.0, which means that an infinite value of SEARCHDIST is used.

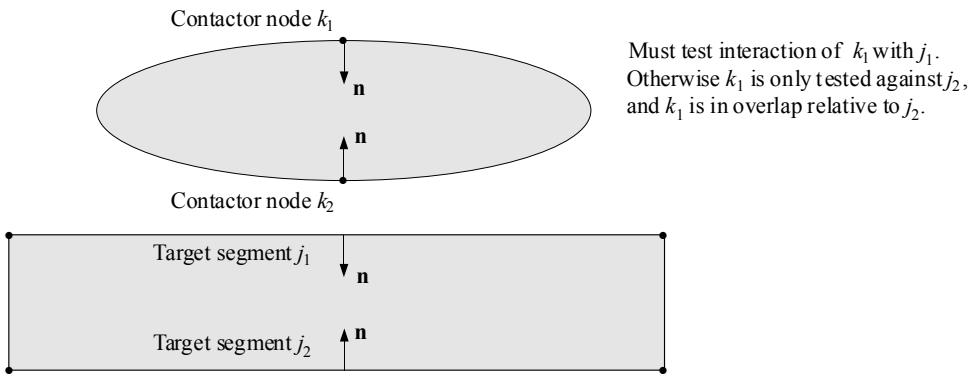
Opposite normals test

For some problems, the Cat 2 contact search can be sped up by applying the opposite normals test (OPPNTEST=YES in the CONTACT-CONTROL and CGROUP commands). The opposite normal test is shown in Fig. 4-11.12.

Because the opposite normals test cannot be used in general, the opposite normals test is not used by default.



a) Example in which opposite normals test can be used, target surface normals all point in the same direction



b) Example in which opposite normals test cannot be used, target surface normals point in opposite directions

Figure 4.11-12: Opposite normals test in Cat 2 contact

4.11.3 Suppression of contact oscillations with NSUPPRESS

In some problems contactor nodes may oscillate during equilibrium iterations between several (usually two) neighboring target segments. Frequently, both solutions are acceptable. A special procedure, termed the NSUPPRESS procedure, can be used to prevent such oscillations. In this case, the program records the pairing target segment for each contactor node in the previous NSUPPRESS iterations. If NSUPPRESS iterations have occurred, and the contactor node is still in contact, and the pairing target segment is one of those recorded in previous iterations, the

suppression feature is activated. The contactor node from this iteration onwards is associated with only that target segment. The contactor node may remain in contact with the segment, or in contact with an infinite plane passing through the segment, or it can separate from contact completely. The contactor node is released from its restrictions once iteration ceases, either because convergence is reached, or due to non-convergence.

If this oscillation suppression feature is used, it is recommended that NSUPPRESS be set greater or equal to 5 and at least 5 less than the maximum number of iterations.

Note that there is memory overhead associated with this feature, where an integer array of size NSUPPRESS is defined for all contactor nodes.

To select the NSUPPRESS feature, specify a non-zero value for the NSUPPRESS parameter in the CONTACT-CONTROL command. The NSUPPRESS feature is not activated by default.

4.12 Small displacement contact

Small displacement contact is selected by setting DISPLACEMENT=SMALL or DISPLACEMENT=SMALL-REBUILD in the CONTACT-CONTROL and CGROUP commands.

If the small displacement contact feature is used, the contactor elements are determined once in the beginning of the analysis, as shown in Figure 4.12-1.

For each contactor element, the local coordinates of the target point and the normal direction are then kept constant.

Small displacement contact is useful when there is very little relative deformation around the contact region and when there is little or no sliding. For such problems, it is much more computationally efficient to perform only one contact search at the beginning of the analysis, rather than repeating the search every iteration. Also, in some cases, convergence can also be slow or unachievable with the large displacement contact algorithm, for

example as nodes oscillate between one target segment and another equally valid neighboring target segment.

Restarts from small displacement to large displacement contact, and vice versa, are allowed.

In Cat 1 contact, the contactor elements are determined

- ▶ At the beginning of the initial analysis
- ▶ When the contact group is born, if the contact group is unborn at the beginning of the analysis
- ▶ At the beginning of every restart analysis

DISPLACEMENT=SMALL and DISPLACEMENT=SMALL-REBUILD are the same for Cat 1 contact.

In Cat 2 contact, the contactor elements are determined

- ▶ At the beginning of the initial analysis
- ▶ When the contact group is born, if the contact group is unborn at the beginning of the analysis
- ▶ At the beginning of every restart analysis, only if DISPLACEMENT=SMALL-REBUILD

Thus, in Cat 2 contact with DISPLACEMENT=SMALL, the same results are obtained whether or not a restart analysis is performed.

In Cat 1 contact, small displacement contact cannot be used in FSI analysis.

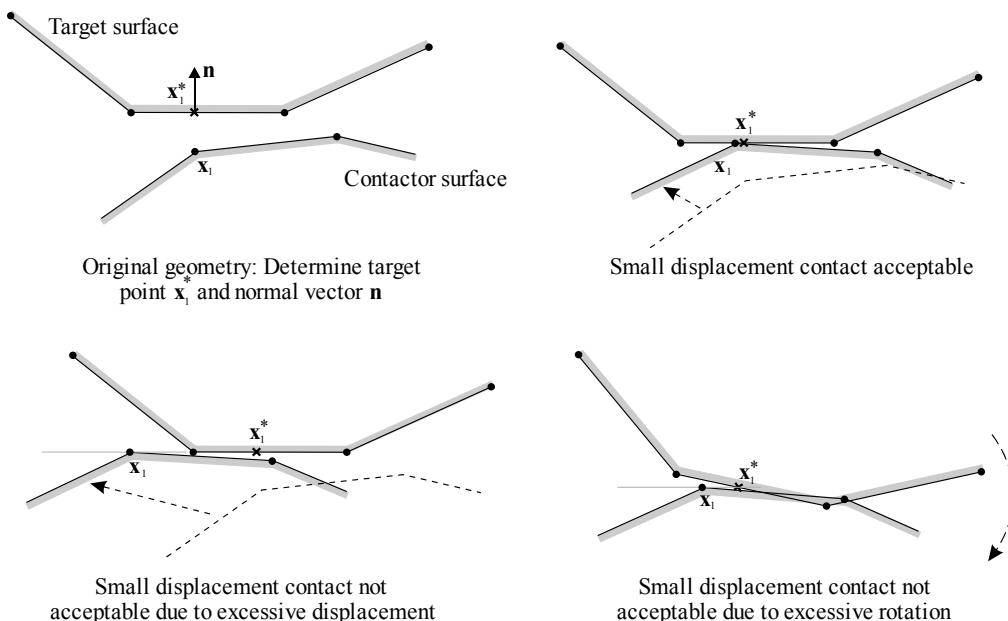


Figure 4.12-1: Small displacement contact feature

4.13 Contactor element normal and tangential directions

The normal direction of the contactor element is obtained from the target segment and isoparametric coordinates on the target segment. If segment normals are selected, the target segment geometry is used to obtain the segment normal. If continuous normals are selected, the nodal point normals on the target segment nodes are interpolated to the target point.

See Section 4.8.1.5 for more information about continuous and segment normals.

When there is friction, the program constructs tangential directions orthogonal to the normal direction.

4.14 Contactor element overlaps and gaps

The gap between the contactor node and target segment is measured in the direction of the contactor element normal (equation (4.2-1) in which \mathbf{n} is the contactor element normal). If offsets are used, the gap is modified by the offset value at the target point.

The overlap is the negative of the gap.

For relative contact damping, the gap velocity is calculated using equation (4.5-1).

4.15 Tied contact

The tied contact feature can be used to bond two contact surfaces together. However, this feature can be used only in Cat 1 contact.

When the tied contact feature is selected for a contact group, ADINA performs an initial contact check at the start of the analysis. All contactor nodes that are found to be in contact or overlapping are permanently attached to their respective target segments. Contactor nodes that are not in contact are also set to be tied if the contact gap is less than a user-specified tied contact tolerance. This tolerance is useful when the contact gap is due to non-matching finite element discretizations of the contact surfaces.

The tied contact feature is conceptually similar to using rigid links or constraints to attach the node to the target surface. The main difference is that the coefficients for the rigid links are automatically determined by the program and they are only applied for the nodes that are initially in contact. The basic idea is illustrated in Fig. 4.15-1.

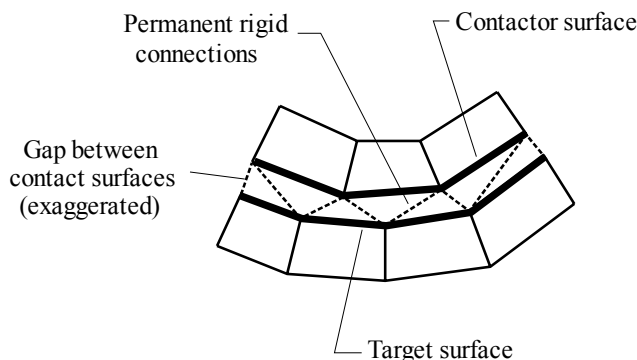


Figure 4.15-1: Tied contact option

Tied contact is not "real" contact because there can be tension between tied contact surfaces. Also no sliding can occur between tied contact surfaces.

The tied contact option can be used to bond two surfaces together if the surfaces have incompatible meshes. However, the surface glueing feature described in Section 5.16 produces more accurate "glue" equations.

If the contact surfaces initially overlap, they are not pushed back to eliminate the overlap. Similarly, if there is an initial gap, it is not eliminated.

The tied contact constraint equations are computed based on the initial nodal positions only. The constraints generated in tied contact are not updated during the analysis. Hence, the constraints will be inaccurate if the bodies experience large rotations.

Tied contact is selected using `CGROUP ... TIED=SMALL`

Due to the limitations of tied contact, glueing should be used instead of tied contact whenever possible.

4.16 Initial penetration features

4.16.1 Introduction

The procedure given in Section 4.14 is used to obtain the geometric overlap between the contactor node and target segment. Using the initial penetration features described in this section, this geometric overlap is modified / adjusted in order to model certain situations.

For example, if an initial overlap is to be ignored, and if, at the start of solution, the geometric overlap at a contactor node is greater than zero, the overlap at the contactor node is reset to zero.

All of the features that modify the geometric gap are termed “initial penetration features”.

Table 4.16.1 summarizes the initial penetration features.

Table 4.16.1. Initial penetration features

Purpose	Option in user interface	Option in command line, CGROUP command
To immediately apply the initial contact conditions	Initial Penetration into Target: Eliminate Time to Eliminate Initial Penetration = 0	INITIAL-PENETRATION=ELIMINATED, TIME-PENETRATION=0 or INITIAL-PENETRATION=PRINT, TIME-PENETRATION=0
To gradually apply the initial contact conditions	Initial Penetration into Target: Eliminate Time to Eliminate Initial Penetration > 0	INITIAL-PENETRATION=ELIMINATED, TIME-PENETRATION>0
To immediately apply an initial overlap or gap	Initial Penetration into Target: Override Time to Eliminate Initial Penetration = 0	INITIAL-PENETRATION=GAP-OVERRIDE TIME-PENETRATION=0
To gradually apply an initial overlap or gap	Initial Penetration into Target: Override Time to Eliminate Initial Penetration > 0	INITIAL-PENETRATION=GAP-OVERRIDE TIME-PENETRATION>0
To ignore the initial overlaps for nodes with positive initial overlap	Initial Penetration into Target: Ignore	INITIAL-PENETRATION=IGNORED

In the command-line, the word DISCARDED is the same as IGNORED, and the word ALLOWED is the same as ELIMINATED. DISCARDED and ALLOWED are used for backwards compatibility with previous versions of ADINA.

For Cat 2 contact, TIME-PENETRATION can be set to FIRST-STEP. In this case TIME-PENETRATION is automatically set to the first user-specified solution time.

The default for Cat 1 contact is to immediately apply the initial contact conditions.

The default for Cat 2 contact is to gradually apply the initial contact conditions with the time to eliminate the initial penetration set to FIRST-STEP.

Figure 4.16-1 illustrates some of the initial penetration features.

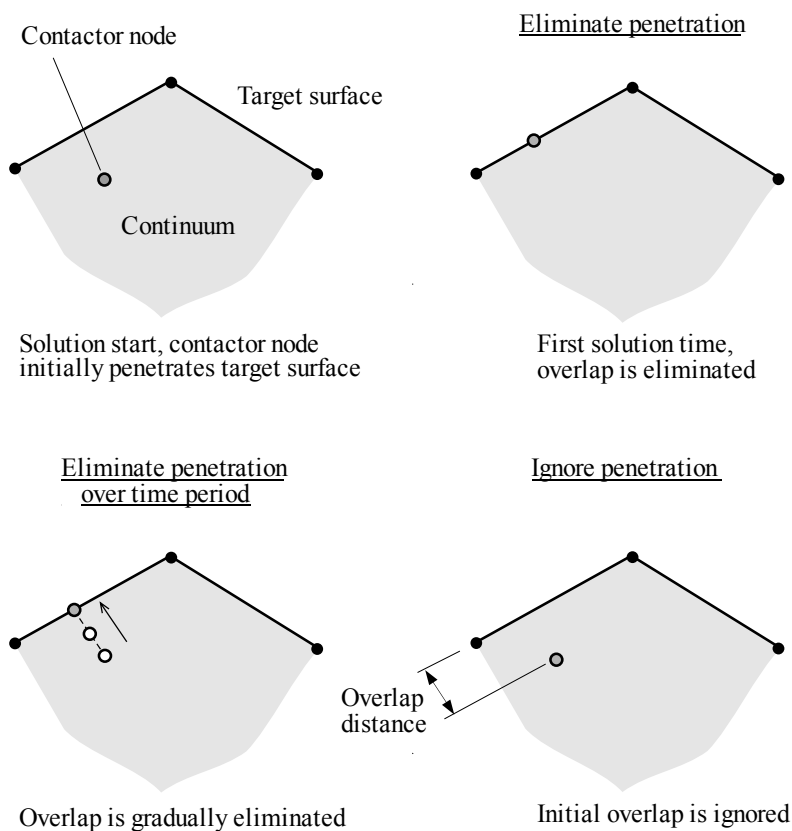


Figure 4.16-1: Initial penetration features

All of the initial penetration features add an additional overlap to the geometric overlap in order to obtain an effective overlap. Then the normal contact forces are generated based on the effective overlap. For example, supposing that the normal contact response gives zero contact force for zero overlap, then if the effective overlap is zero, the contact force is zero.

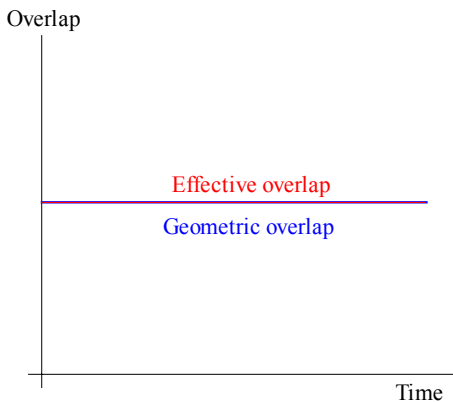
In the following, we consider a single contactor node, and discuss how the additional overlap is obtained.

4.16.2 Eliminate: Immediately apply the initial contact conditions

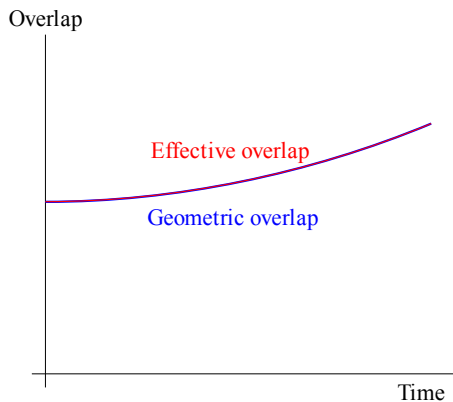
When the initial contact conditions are to be immediately applied, the additional overlap is zero and the effective overlap is equal to the geometric overlap.

If the geometric overlap happens to be positive, the normal contact forces will tend to eliminate the overlap, in the absence of additional forces acting on the contactor node.

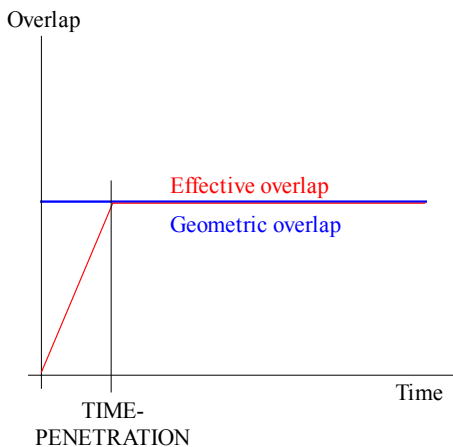
Figs. 4.16-2(a) and 4.16-2(b) show the geometric and effective overlaps when the eliminate option is used with zero time-penetration.



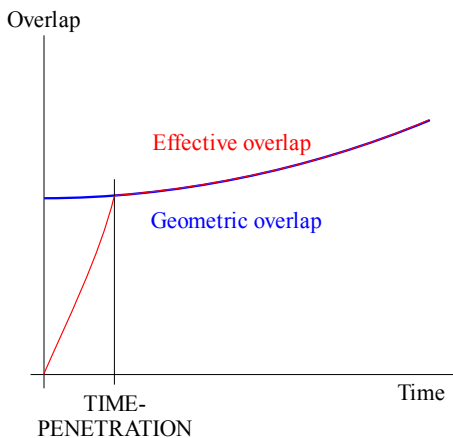
(a) Effective overlap used with eliminate option,
TIME-PENETRATION=0,
geometric overlap not changing with time



(b) Effective overlap used with eliminate option,
TIME-PENETRATION=0,
geometric overlap changing with time



(c) Effective overlap used with eliminate option,
TIME-PENETRATION>0,
geometric overlap not changing with time



(d) Effective overlap used with eliminate option,
TIME-PENETRATION>0,
geometric overlap changing with time

Figure 4.16-2: Geometric and effective overlaps used with eliminate option

4.16.3 Eliminate: Gradually apply the initial contact conditions

When the initial contact conditions are to be gradually applied, the initial nodal overlap is recorded. If the overlap is negative, the additional overlap is zero and the effective overlap is negative. If the overlap is positive, the additional overlap is initially set to the negative of the initial nodal overlap, with the result that, initially, the effective overlap is zero. Thus, initially, the effective overlap is never positive.

During the time interval between the time of solution start and the time TIME-PENETRATION, the additional overlap is gradually reduced to zero, thus the effective overlap approaches the geometric overlap. For solution times greater than TIME-PENETRATION, the additional overlap is set to zero and the effective overlap equals the geometric overlap.

This feature is useful when initially some of the overlaps are large. The large initial overlaps might make it difficult for the model to converge in the first solution step.

When TIME-PENETRATION=FIRST-STEP, then, if there are no ATS cutbacks, the solution is obtained at the first user-specified solution time, the additional overlap is zero and the same solution is obtained as if TIME-PENETRATION is set to zero. However, if there are ATS cutbacks, then during the ATS sub-steps, the additional overlap is non-zero so that overlaps are temporarily ignored.

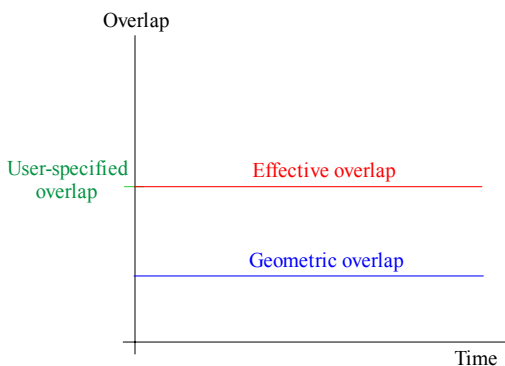
Figs. 4.16-2(c) and 4.16-2(d) show the geometric and effective overlaps when the eliminate option is used with non-zero time-penetration.

4.16.4 Gap-override: Immediately apply an initial overlap or gap

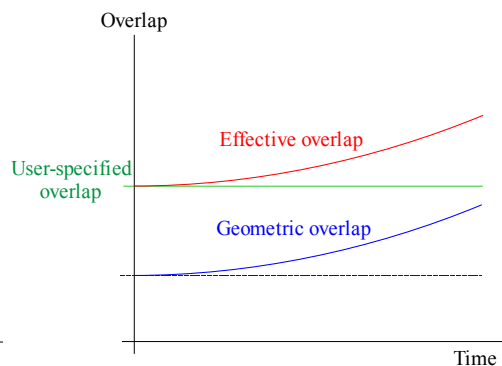
At the start of solution, the additional overlap is computed so as to set the effective overlap equal to the user-specified initial overlap.

Notice that the user input is the user-specified **gap**. The user-specified gap can be set negative to specify a user-specified overlap.

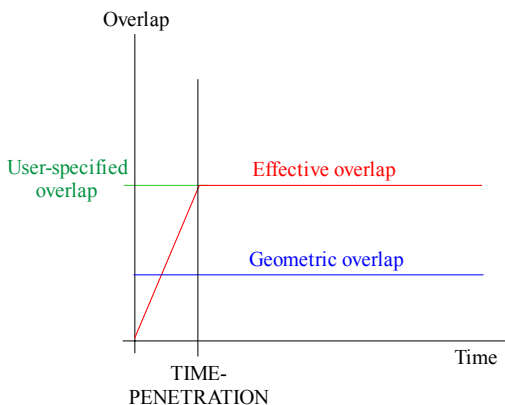
Figs. 4.16-3(a) and 4.16-3(b) show the geometric and effective overlaps when the gap-override option is used with zero time-penetration.



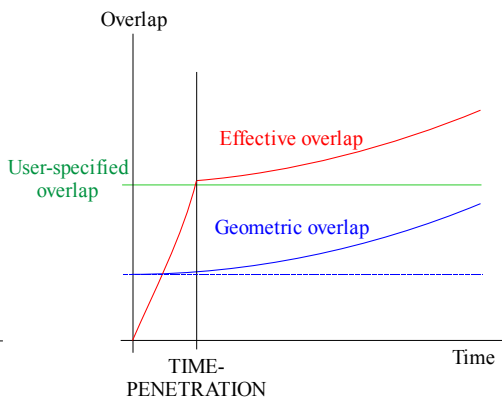
(a) Effective overlap used with gap-override option, $\text{TIME-PENETRATION}=0$, geometric overlap not changing with time



(b) Effective overlap used with gap-override option, $\text{TIME-PENETRATION}=0$, geometric overlap changing with time



(c) Effective overlap used with gap-override option, $\text{TIME-PENETRATION}>0$, geometric overlap not changing with time



(d) Effective overlap used with gap-override option, $\text{TIME-PENETRATION}>0$, geometric overlap changing with time

Figure 4.16-3: Geometric and effective overlaps used with gap-override option

The gap-override feature is useful for problems involving curved meshes in close proximity, such as the shrink fit example shown in Fig. 4.16-4. The gaps and penetrations measured from the discretized finite element mesh are sometimes inaccurate for such problems (unless matching meshes are used). In some problems, such as that shown in the figure, a constant geometry based overlap should be applied to all nodes, which corresponds to a gap override value of $-\delta$.

Note that mesh refinement and quadratic elements reduce the error in the measured overlaps but frequently a very high mesh density would have to be used if gap override is not used.

Note also that the error in mesh based gaps and penetrations for curved surfaces can be more significant when low precision numbers are used for the node coordinates. Gap override is also useful for such cases.

Two rings with a geometric overlap δ (shrink fit)

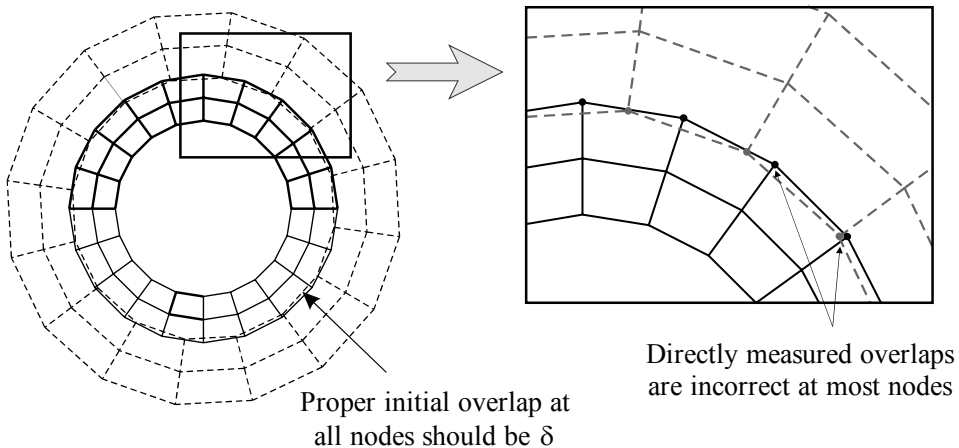


Figure 4.16-4: Significance of gap override for curved non-matched geometries

4.16.5 Gap-override: Gradually apply an initial overlap or gap

At the start of solution, the additional overlap is computed so as to set the effective overlap equal to zero.

During the time interval between the time of solution start and the time TIME-PENETRATION, the additional overlap is gradually adjusted, so that at solution time TIME-PENETRATION, the additional overlap is set as follows: if the geometric overlap equals the initial overlap, the effective overlap equals the user-specified overlap.

Figs. 4.16-3(c) and 4.16-3(d) show the geometric and effective overlaps when the eliminate option is used with non-zero time-penetration.

4.16.6 Ignore: Ignoring the initial overlap for nodes with positive overlap

If the node has an initial positive overlap, the additional overlap is computed so as to set the initial effective overlap equal to zero.

Fig. 4.16-5 shows the geometric and effective overlaps when the ignore option is used.

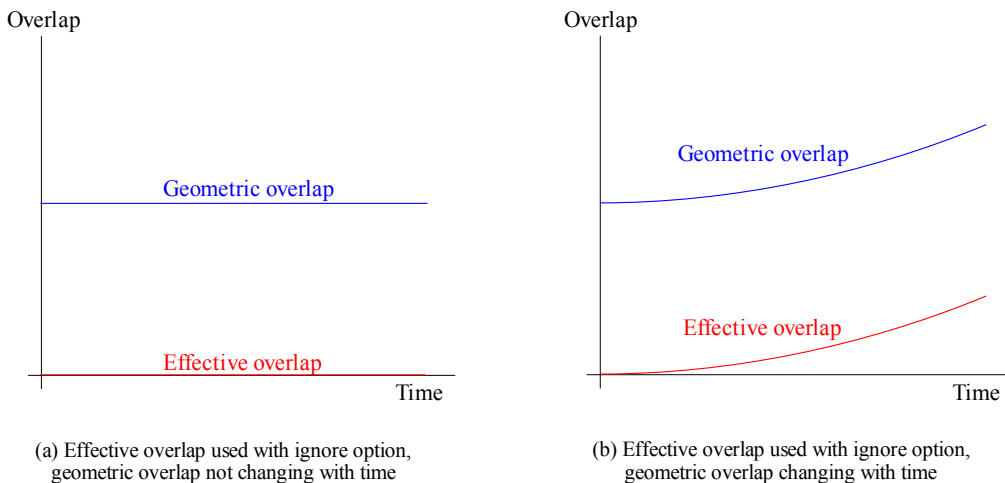


Figure 4.16-5: Geometric and effective overlaps used with ignore option

Ignoring initial overlaps is a useful option when these overlaps are just a product of the finite element discretization, meaning that they do not exist in the physical model. Fig. 4.16-6 illustrates one such case involving contact between concentric cylinders. In this situation, if initial overlaps are not ignored, the contact algorithm will try to push the overlapping contactor nodes to the target surface segments in the first step, creating initial prestressing. These initial overlaps and any prestressing that they might cause are unrealistic. Ignoring the initial overlaps is useful in this case. Note however, that if either cylinder is significantly rotated the initial overlaps calculated at each contactor node (in the initial configuration) will no longer be valid. In this case, the best alternative would be to use a much finer mesh.

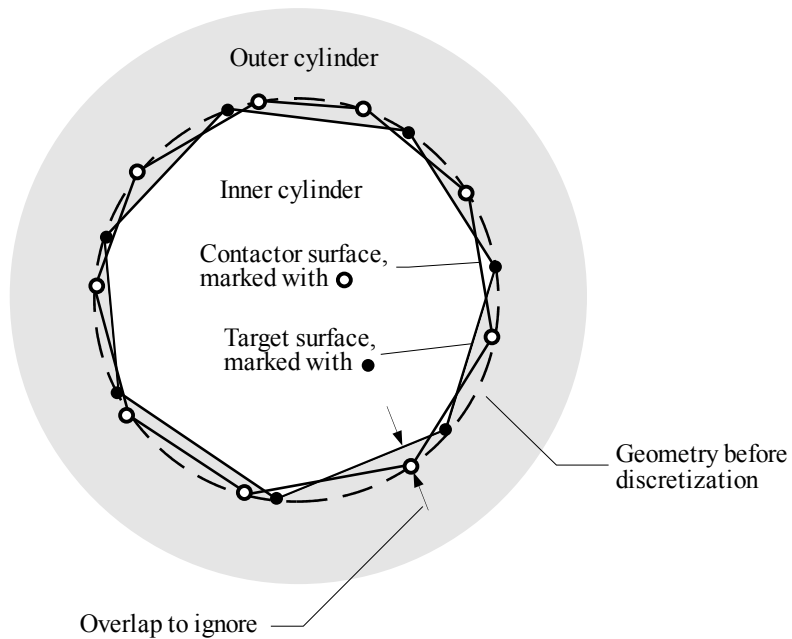


Figure 4.16-6: Analysis of contact between concentric cylinders, initial penetration is ignored

4.17 Contactor element relative velocities

4.17.1 Gap velocity

The gap velocity used in relative contact damping is evaluated as described in Section 4.5.1.

The contactor node and target segment velocities are evaluated as the change in displacements divided by the change in solution time. In this way the same gap velocity evaluation is used in static and dynamic analysis.

4.17.2 Slip velocities

The slip velocity at a contactor node is defined as the velocity of the contactor node relative to the velocity of the contacting point of the target surface, projected onto the tangential plane, see Section 4.3.2.

The contactor node and target segment velocities are evaluated as the change in displacements divided by the change in solution time. In this way the same slip velocity evaluation is used in static and dynamic analysis.

If contact slip velocity loads are used, then the velocities of the contactor node and target segment include the slip velocity loads, see the discussion of contact slip velocity loads in Section 4.8.1.13.

4.17.3 Sliding distances

The sliding distance is computed as the time integral of the slip velocity magnitude.

4.18 Contactor element equivalent areas

4.18.1 Definition of the equivalent area

Each contactor node k has an equivalent area A_k . The equivalent area is used to compute contact tractions from contact forces, e.g. $p_g = \lambda_g / A_k$. In Cat 2 contact, the equivalent area is also used to compute contact forces from contact tractions, e.g. $\lambda_g = p_g A_k$.

Supposing that the contact traction is uniform over a contact surface, then the “correct” formula for the contactor node equivalent area is

$$A_k = \int h_k dA \quad (4.18-1)$$

where h_k is the contactor node shape function and the integral is taken over all contactor segments attached to the contactor node.

For 2-D axisymmetric contact segments, the differential segment area is $dA = r d\ell$ in which $d\ell$ is the differential segment length and r is the distance to the centerline. This definition is consistent with the “1 radian” assumption used in the axisymmetric elements.

For 2-D plane stress contact segments, the differential segment area is $dA = b d\ell$ in which $d\ell$ is the differential segment length and b is the out-of-plane thickness. For Cat 1 contact, unit thickness is always assumed ($b = 1$). For Cat 2 contact, the thickness can be entered using the THICKNESS parameter of the CGROUP command.

For 2-D plane strain contact segments, the differential segment area is $dA = d\ell$, that is, unit thickness is assumed. This definition is consistent with the “unit thickness” assumption used in the plane strain elements.

The reason that formula (4.18-1) is correct is that then the consistent nodal force corresponding to the (constant) contact pressure is then

$$F_k = p \int h_k dA = pA_k \quad (4.18-2)$$

This same formula is used, for example, in the calculation of consistent nodal forces corresponding to applied pressures.

4.18.2 Equivalent areas for higher-order contact segments

However, in contact, the formula $A_k = \int h_k dA$ cannot generally be used for the calculation of the contact node equivalent area. This is because the value of $\int h_k dA$ might be zero or negative for higher-order contact segments (contact segments with mid-side or mid-face nodes). The non-positive value conflicts with the requirement that the normal contact force act in the same direction as the contact pressure.

As a specific example, consider frictionless contact between a rigid target and an element with uniform compressive stress (Fig 4.18-1). In order that equilibrium be satisfied, the contactor node forces must be equal to the element nodal point forces. If the element is a rectangular 20-node brick element, then the element nodal point forces are tensile at the corner nodes and compressive at the mid-side nodes. The contact tractions are clearly compressive at all of the contact segment nodes.

In order to obtain the expected results for this problem, it is necessary to allow the equivalent areas at the corner nodes to be negative.

Now consider the same problem, but with a large friction coefficient and a nonzero Poisson's ratio. Since there is friction, tangential forces will act on the contact nodes to resist the Poisson ratio expansion, as shown in Fig. 4.18-2.

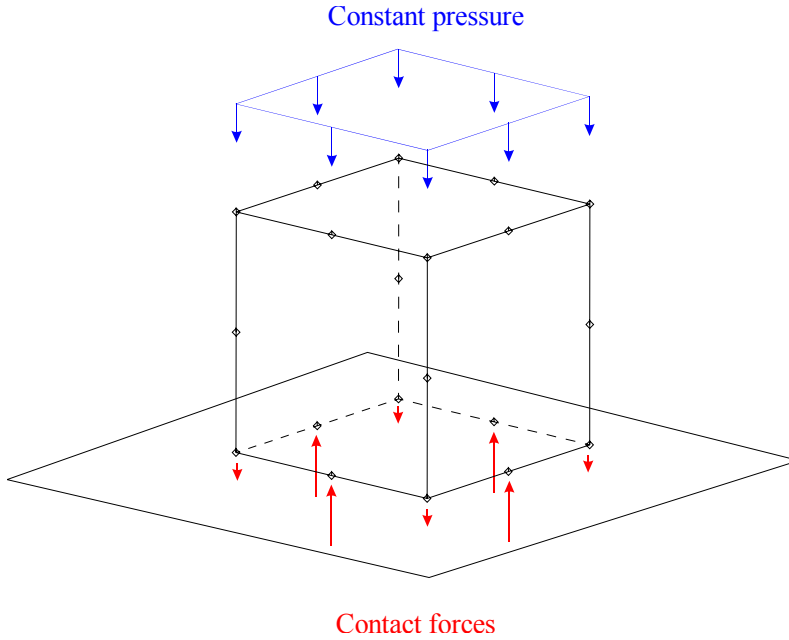


Figure 4.18-1: Frictionless contact between a 20-node brick element and rigid target

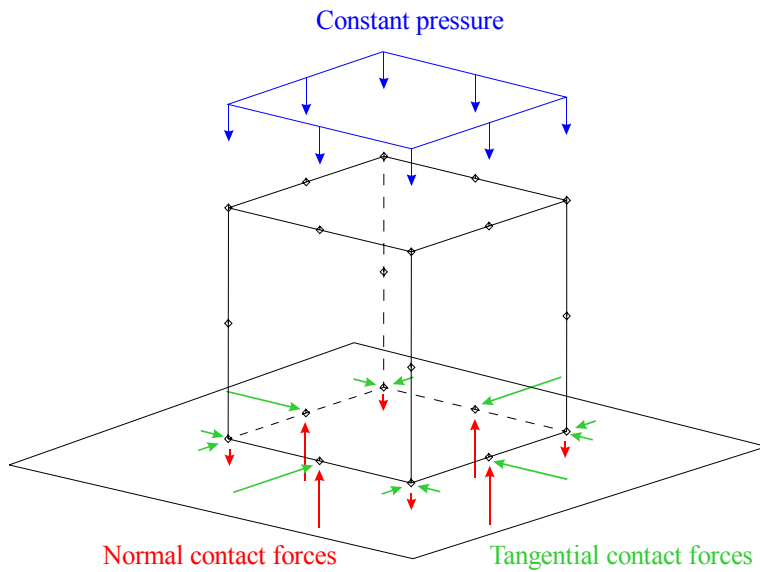


Figure 4.18-2: Frictional contact between a 20-node brick element and rigid target

In each of the tangential directions, the tangential contact forces on the corner nodes act in the same direction as the tangential contact forces on the mid-side nodes.

If negative equivalent areas are used at the corner nodes, the wrong results are obtained for the tangential forces.

For general contact conditions, we therefore by default use equivalent areas that are positive, even though these positive equivalent areas give the wrong results for problems in which the contact tractions are constant over the contact surface. Special procedures are available so that the correct results are obtained for problems in which the contact tractions are constant over the contact surface, see below.

For reference, here is a table of 3-D higher-order contact segments and the elements to which they are typically attached.

Contact segment	Typically attached to	Comments
6-node segment	10- or 11-node tetrahedral element	$\int h_k dA = 0$ for the corner nodes of an equilateral segment.
8-node segment	20- or 21-node brick element	$\int h_k dA < 0$ for the corner nodes of a rectangular segment.
9-node segment	27-node brick element	$\int h_k dA > 0$ for the corner nodes of a rectangular segment

Notice that the 9-node rectangular segment has positive equivalent area at all nodes, including the corner nodes. Therefore the 27-node brick element should be used instead of the 20- or 21-node brick element whenever possible.

In 2D axisymmetric analysis with 3-node target segments, if the corner node is on the axisymmetric centerline and the other nodes on the segment are not on the axisymmetric centerline, then the corner node on the axisymmetric centerline has zero equivalent area.

4.18.3 Equivalent areas for Cat 1 contact

For Cat 1 contact, the formula $A_k = \int h_k dA$ is used for all lower-order contactor segments. A modification to this formula is used for the 6- and 8-node higher-order segments, such that $A_k > 0$ for these segments, and such that the sum of the contactor node areas equals the contact surface area.

4.18.4 Tension-consistent feature for Cat 1 contact

In order to obtain the expected results for frictionless problems with uniform contact traction and higher-order elements, the program can accept tensile forces as if they are compressive. This option is selected by setting TENSION-CONSISTENT=YES in the CONTACT-CONTROL command. Accepting these tensile forces gives the expected results.

However, for more general contact conditions, in which the contact tractions are not uniform, the TENSION-CONSISTENT feature may slow down or even prevent convergence.

The TENSION-CONSISTENT feature is off by default.

4.18.5 Equivalent areas for Cat 2 contact

For Cat 2 contact, the formula $A_k = \int h_k dA$ is used for all lower-order contactor segments.

For the higher-order contact segments, the larger of $A_k = \int h_k dA$ and $A_k = \int h_k^2 dA$ is used for the corner nodes, and the remaining contact areas are adjusted in order that the sum of the contactor node areas equals the contact surface area. This formula yields $A_k = \int h_k dA$ for all nodes of a 9-node rectangular segment, therefore the equivalent areas are not modified for 9-node rectangular segments.

4.18.6 Tension-consistent feature for Cat 2 contact

In order to obtain the expected results for frictionless problems with uniform contact traction and higher-order elements, the program optionally can always use the formula $A_k = \int h_k dA$ for all nodes, regardless if A_k is positive, zero or negative.

This option is selected by setting TENSION-CONSISTENT=YES in the CONTACT-CONTROL command. The expected results are obtained for frictionless problems with uniform contact traction and higher-order elements.

However, for more general contact conditions, in which the contact tractions are not uniform, the TENSION-CONSISTENT feature may slow down or even prevent convergence. This is especially true for problems in which only the corner nodes of higher-order segments (and not the mid-side nodes) are in contact.

The TENSION-CONSISTENT feature is off by default.

Because the TENSION-CONSISTENT feature is implemented differently for Cat 1 and Cat 2 contact, problems that work in Cat 1 contact with TENSION-CONSISTENT=YES might not work in Cat 2 contact with TENSION-CONSISTENT=YES.

4.18.7 Equivalent areas for contact bodies

The contact equivalent areas for contactor nodes on contact bodies is $A_k = 1$.

4.19 Contactor elements in implicit analysis

4.19.1 The constraint function method

In implicit analysis, the contact conditions are included by the use of Lagrange multipliers and constraint functions, as described in detail below.

Each contactor node has Lagrange multiplier degrees of freedom in addition to the usual displacement degrees of freedom.

Corresponding to the additional Lagrange multiplier unknowns are constraint functions that couple the additional unknowns to the motions of the contact surfaces. During the solution process, the program attempts to set the values of all of the constraint functions to zero.

The normal contact response is included using a normal Lagrange multiplier and normal constraint function, and the frictional contact response is included using frictional Lagrange multipliers and frictional constraint functions.

4.19.1.1 The constraint function method in Cat 1 contact

Consider a single 2-D contactor element with contactor node and two-node target segment.

For frictionless contact, the system of equations for this contactor element has the form

$$\begin{bmatrix} * & 0 & 0 & * \\ 0 & * & * & * \\ 0 & * & * & * \\ * & * & * & * \end{bmatrix} \begin{bmatrix} \Delta u_n^C \\ \Delta u_n^{T_1} \\ \Delta u_n^{T_2} \\ \Delta \lambda_g \end{bmatrix} = \begin{bmatrix} * \\ * \\ * \\ 0 \end{bmatrix} - \begin{bmatrix} -\lambda_g \\ h_1 \lambda_g \\ h_2 \lambda_g \\ w(g, \lambda_g) \end{bmatrix} \quad (4.19-1)$$

in which u_n^C is the displacement of the contactor node, measured in the direction of the target normal, $u_n^{T_1}$, $u_n^{T_2}$ are the displacements of the target nodes, measured in the direction of the target normal, h_1 , h_2 are the shape function values at the contacting point on the target segment, λ_g is the Lagrange multiplier, representing a contact force, and $w(g, \lambda_g)$ is the constraint function. Observe that the Lagrange multiplier provides an additional term in the right-hand-side of the system of equations.

As described in Chapter 7, the program performs equilibrium iterations to obtain a converged solution, in which the right-hand side of the system of equations is zero. Since the term $0 - w(g, \lambda_g)$ appears on the right-hand-side, at convergence, $w(g, \lambda_g) = 0$, so that the constraint function is satisfied.

Also observe that the contactor and target nodes are connected on the left-hand-side only by the λ_g contact equations.

For frictional contact, the system of equations has the form

$$\begin{bmatrix}
 * & * & 0 & 0 & 0 & 0 & * & 0 \\
 * & * & 0 & 0 & 0 & 0 & 0 & * \\
 0 & 0 & * & * & * & * & * & 0 \\
 0 & 0 & * & * & * & * & 0 & * \\
 0 & 0 & * & * & * & * & * & 0 \\
 0 & 0 & * & * & * & * & 0 & * \\
 \hline
 * & 0 & * & 0 & * & 0 & * & 0 \\
 0 & * & 0 & * & 0 & * & * & *
 \end{bmatrix}
 \begin{bmatrix}
 \Delta u_n^C \\
 \Delta u_t^C \\
 \Delta u_n^{T_1} \\
 \Delta u_t^{T_1} \\
 \Delta u_n^{T_2} \\
 \Delta u_t^{T_2} \\
 \hline
 \Delta \lambda_g \\
 \Delta \tau_t
 \end{bmatrix}
 =
 \begin{bmatrix}
 * \\
 * \\
 * \\
 * \\
 * \\
 * \\
 \hline
 0 \\
 0
 \end{bmatrix}
 -
 \begin{bmatrix}
 -\lambda_g \\
 -\tau_t \\
 h_1 \lambda_g \\
 h_1 \tau_t \\
 h_2 \lambda_g \\
 h_2 \tau_t \\
 \hline
 w(g, \lambda_g) \\
 v(\dot{u}, \tau_t, \lambda_g)
 \end{bmatrix} \quad (4.19-2)$$

in which u_t^C is the displacement of the contactor node, measured in the direction of the target tangential direction, $u_t^{T_1}$, $u_t^{T_2}$ are the displacements of the target nodes, measured in the direction of the target tangential direction, τ_t is the frictional Lagrange multiplier, representing a frictional contact force, \dot{u} is the slip velocity magnitude and $v(\dot{u}, \tau_t, \lambda_g)$ is the frictional constraint function. At convergence, both $w(g, \lambda_g) = 0$ and $v(\dot{u}, \tau_t, \lambda_g) = 0$.

4.19.1.2 The constraint function method in Cat 2 contact

The constraint function method in Cat 2 contact is similar to the constraint function method in Cat 1 contact. The primary difference is that, in Cat 2 contact, the Lagrange multipliers represent tractions instead of forces. Therefore the equivalent areas

of the contactor nodes appear on the right-hand-side of the system of equations.

Consider a single 2-D contactor element with contactor node and two-node target segment.

For frictionless contact, the system of equations for this contactor element has the form

$$\begin{bmatrix} * & 0 & 0 & * \\ 0 & * & * & * \\ 0 & * & * & * \\ * & * & * & * \end{bmatrix} \begin{bmatrix} \Delta u_n^C \\ \Delta u_n^{T_1} \\ \Delta u_n^{T_2} \\ \Delta p_g \end{bmatrix} = \begin{bmatrix} * \\ * \\ * \\ 0 \end{bmatrix} - \begin{bmatrix} -Ap_g \\ h_1 Ap_g \\ h_2 Ap_g \\ w(g, p_g) \end{bmatrix} \quad (4.19-3)$$

in which p_g is the Lagrange multiplier, representing a contact traction, A is the equivalent area of the contactor node, $w(g, p_g)$ is the constraint function and the other quantities are the same as in Cat 1 contact.

For frictional contact, the system of equations has the form

$$\begin{bmatrix} * & * & 0 & 0 & 0 & 0 & * & 0 \\ * & * & 0 & 0 & 0 & 0 & 0 & * \\ 0 & 0 & * & * & * & * & * & 0 \\ 0 & 0 & * & * & * & * & 0 & * \\ 0 & 0 & * & * & * & * & * & 0 \\ 0 & 0 & * & * & * & * & 0 & * \\ * & 0 & * & 0 & * & 0 & * & 0 \\ 0 & * & 0 & * & 0 & * & * & * \end{bmatrix} \begin{bmatrix} \Delta u_n^C \\ \Delta u_t^C \\ \Delta u_n^{T_1} \\ \Delta u_t^{T_1} \\ \Delta u_n^{T_2} \\ \Delta u_t^{T_2} \\ \Delta p_g \\ \Delta T_t \end{bmatrix} = \begin{bmatrix} * \\ * \\ * \\ * \\ * \\ * \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} -Ap_g \\ -AT_t \\ h_1 Ap_g \\ h_1 AT_t \\ h_2 Ap_g \\ h_2 AT_t \\ w(g, p_g) \\ v(\dot{u}, T_t, p_g) \end{bmatrix} \quad (4.19-4)$$

in which T_t is the frictional Lagrange multiplier, representing a frictional contact traction, $v(\dot{u}, T_t, p_g)$ is the frictional constraint function and the other quantities are the same as in Cat 1 contact.

4.19.1.3 Many nodes in contact

When considering many nodes in contact, the system of equations has the structure

$$\left[\begin{array}{ccc|c} \mathbf{KUU} & \mathbf{KUV} & \mathbf{KUW} & \mathbf{0} \\ \mathbf{KVU} & \mathbf{KVV} & \mathbf{0} & \mathbf{KVA} \\ \mathbf{KWU} & \mathbf{0} & \mathbf{KWW} & \mathbf{KWA} \\ \hline \mathbf{0} & \mathbf{KAV} & \mathbf{KAW} & \mathbf{KAA} \end{array} \right] \begin{bmatrix} \Delta \mathbf{U} \\ \Delta \mathbf{V} \\ \Delta \mathbf{W} \\ \Delta \mathbf{A} \end{bmatrix} = \begin{bmatrix} \mathbf{RU} - \mathbf{FU} \\ \mathbf{RV} + \mathbf{FA} - \mathbf{FV} \\ \mathbf{RW} + \mathbf{HA} - \mathbf{FW} \\ \hline -\mathbf{w}(\mathbf{V}, \mathbf{W}, \mathbf{A}) \end{bmatrix} \quad (4.19-5)$$

in which

\mathbf{U} = displacement degrees of freedom not on contactor or target surfaces

\mathbf{V} = displacement degrees of freedom of nodes on contactor surfaces

\mathbf{W} = displacement degrees of freedom of nodes on target surfaces

\mathbf{A} = Lagrange multipliers, both normal and frictional. In Cat 1 contact, the Lagrange multipliers are forces, in Cat 2 contact, the Lagrange multipliers are tractions.

\mathbf{K}^{**} = stiffness matrix coupling degrees of freedom, for example \mathbf{KUV} connects the \mathbf{U} and \mathbf{V} degrees of freedom.

\mathbf{F}^* = internal force vector for degrees of freedom, for example \mathbf{FU} is the internal force vector for the \mathbf{U} degrees of freedom

\mathbf{R}^* = external force vector for degrees of freedom, for example \mathbf{RU} is the external force vector for the \mathbf{U} degrees of freedom

\mathbf{FA} = contact forces from Lagrange multipliers acting onto contactor degrees of freedom.

$\mathbf{H}\mathbf{A}$ = contact forces from Lagrange multiplier forces acting onto target degrees of freedom

$\mathbf{w}(\mathbf{V}, \mathbf{W}, \mathbf{\Lambda})$ = constraint functions, both normal and frictional.

The matrices \mathbf{KVA} , \mathbf{KWA} , \mathbf{KAV} , \mathbf{KAW} , \mathbf{KAA} are constructed by the contact algorithm. These matrices change their structure as the contactor elements change, for example, when contactor nodes enter contact, or when contactor nodes change target segments. Thus the bandwidth and sparse structure of the global stiffness matrix also changes as the contactor elements change.

When a symmetric solver is used, the contact matrices \mathbf{KVA} , \mathbf{KWA} , \mathbf{KAV} , \mathbf{KAW} , \mathbf{KAA} are also symmetrized, that is $\mathbf{KAV} = \mathbf{KVA}$, etc.

4.19.1.4 Overconstraining

In the constraint function method, it is possible for a contactor node to be overconstrained. To illustrate, we consider a model with two contactor nodes contacting a rigid target under frictionless conditions. The system of equations has the form

$$\begin{bmatrix} * & * & * & 0 \\ * & * & 0 & * \\ * & 0 & \varepsilon & 0 \\ 0 & * & 0 & \varepsilon \end{bmatrix} \begin{bmatrix} \Delta u_1 \\ \Delta u_2 \\ \Delta \lambda_{g1} \\ \Delta \lambda_{g2} \end{bmatrix} = \begin{bmatrix} * \\ * \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} -\lambda_{g1} \\ -\lambda_{g2} \\ w_1(g_1, \lambda_{g1}) \\ w_2(g_2, \lambda_{g2}) \end{bmatrix} \quad (4.19-6)$$

Here ε is related to the compliance of the force-overlap curve. As the contact becomes more and more rigid, $\varepsilon \rightarrow 0$.

This system of equations is well-posed, even if $\varepsilon = 0$, because the third and fourth columns are different.

Now suppose that the two contactor nodes are constrained to each other, so that $\Delta u_1 = \Delta u_2$. The system of equations now has the form

$$\begin{bmatrix} * & * & * \\ * & \varepsilon & 0 \\ * & 0 & \varepsilon \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta \lambda_{g1} \\ \Delta \lambda_{g2} \end{bmatrix} = \begin{bmatrix} * \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} -(\lambda_{g1} + \lambda_{g2}) \\ w_1(g_1, \lambda_{g1}) \\ w_2(g_2, \lambda_{g2}) \end{bmatrix} \quad (4.19-7)$$

As long as the contact is compliant, that is, if ε is large, the system of equations is still well-posed, because the second and third columns are still different. But if the contact is rigid, that is, if $\varepsilon = 0$, the system of equations is singular because the second and third columns are the same.

For $\varepsilon \rightarrow 0$, the system of equations is almost singular and round-off will cause the equation solver to give incorrect results.

Other cases in which this overconstraining of rigid contact can occur is if a contactor node is in two contact groups and is in contact in both groups, or if a contactor node is in a symmetric contact pair.

If the overconstraining cannot be avoided, then it is necessary to use compliant contact.

Although this example assumes Cat 1 contact, the same considerations apply in Cat 2 contact.

4.19.2 Normal pressures and forces for no-overlap contact

4.19.2.1 Gap force vs gap relationship for Cat 1 contact

The Lagrange multiplier for normal contact response represents the normal contact force λ_g acting onto the contactor node.

The normal constraint function is

$$w(g, \lambda_g) = \frac{g + \lambda_g}{2} - \sqrt{\left(\frac{g - \lambda_g}{2}\right)^2 + \varepsilon_N} \quad (4.19-8)$$

where ε_N is a small user-defined parameter (Fig 4.19-1). The curve corresponding to zero constraint function is

$$\lambda_g = \frac{\varepsilon_N}{g} \quad (4.19-9)$$

which is the same as equation (4.2-3) in Section 4.2.2.1. The default value of $\varepsilon_N=1\text{E-}12$ is suitable for most applications and should rarely be modified.

It is possible to set EPSN=0.0. In this case ADINA automatically determines EPSN. However, this determination may not result in correct results for some problems. Hence EPSN=0.0 should not be used in general.

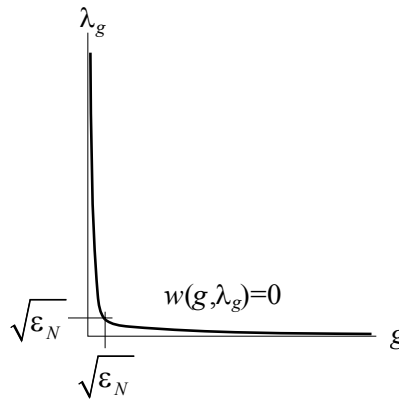


Figure 4.19-1: Constraint function for normal contact, Cat 1 contact

The gap pressure is not used for no-overlap contact in Cat 1 contact.

4.19.2.2 Gap pressure vs gap relationship for Cat 2 contact

The Lagrange multiplier for normal contact response represents the contact gap pressure p_g acting onto the contactor node.

ADINA Structures — Theory and Modeling Guide

$$w(g, p_g) = \frac{g + p_g}{2} - \sqrt{\left(\frac{g - p_g}{2}\right)^2} + \varepsilon_N \quad (4.19-10)$$

where ε_N is a small user-defined parameter (Fig 4.19-2).

The curve corresponding to zero constraint function is

$$p_g = \frac{\varepsilon_N}{g} \quad (4.19-11)$$

which is the same as equation (4.2-4) in Section 4.2.2.1. The default value of $\varepsilon_N=1\text{E-}12$ is suitable for most applications and should rarely be modified.

For Cat 2 contact, EPSN=0.0 is the same as EPSN=1E-12.

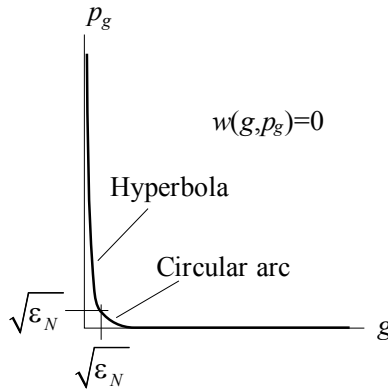


Figure 4.19-2: Constraint function for normal contact, Cat 2 contact

The gap force is computed from the gap pressure using $\lambda_g = A p_g$ where A is the contactor node equivalent area.

4.19.3 Normal pressures and forces for compliant contact

4.19.3.1 Gap force vs gap relationships for Cat 1 contact

The Lagrange multiplier for normal contact response represents the normal contact force λ_g acting onto the contactor node.

The constraint function used for normal contact is modified to include the compliance, as shown in Figure 4.19-3.

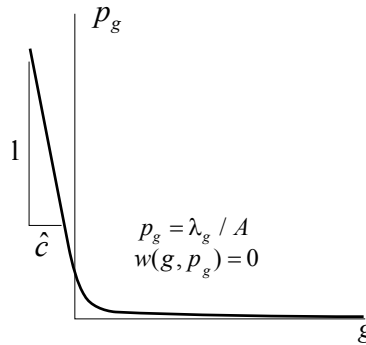


Figure 4.19-3: Compliant constraint function for normal contact, Cat 1 contact

Notice that the contactor node area is included, so that, for large overlap, the relationship between gap and gap pressure is

$$p_g = \frac{1}{\hat{c}} \times (-g) \quad (4.19-12)$$

which is the same as equation (4.2-5) given in Section 4.2.2.2.

4.19.3.2 Gap pressure vs gap relationships for Cat 2 contact

The Lagrange multiplier for normal contact response represents the normal contact pressure p_g acting onto the contactor node.

Constant contact compliance

The relationship between gap and gap pressure, for positive overlap, is

$$p_g = \frac{1}{\hat{c}} \times (-g) \quad (4.19-13)$$

which is the same as equation (4.2-5) given in Section 4.2.2.2. Figure 4.19-4 shows the zero of the constraint function.

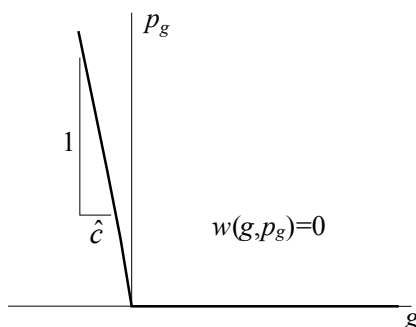


Figure 4.19-4: Compliant constraint function for normal contact, Cat 2 contact

The gap force is computed from the gap pressure using $\lambda_g = A p_g$ where A is the contactor node equivalent area.

Constant contact compliance with backstop

Power law contact pressure

XEXP contact pressure

Multilinear contact pressure

The relationships between gap and gap pressure are given in Sections 4.2.2.3 to 4.2.2.5. Each of these relationships is expressed as the zero of a constraint function, similar to the constraint function used for constant contact compliance.

4.19.4 Relative contact damping pressures and forces, Cat 2 contact

The Lagrange multiplier for normal contact response represents the total contact pressure $p_{total} = p_g + p_{reldamp}$ acting onto the contactor node.

The constraint functions used for normal contact are modified to include the total contact pressure, as shown in Fig. 4.19-5 for the case of constant contact compliance. See Section 4.5 for the formulas used to calculate relative contact damping pressures.

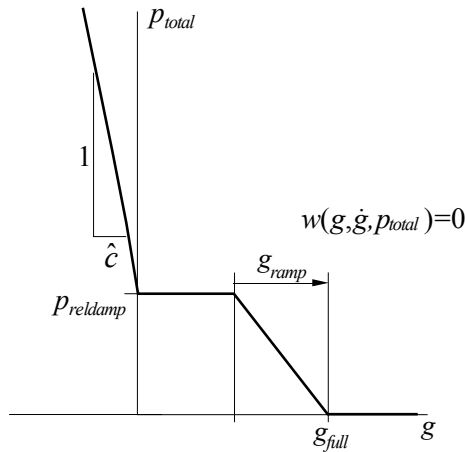


Figure 4.19-5: Normal contact constraint function including relative contact damping constraint function, constant contact compliance

The gap force is computed from the gap pressure using $\lambda_g = A p_g$, and the damping force is computed from the damping pressure using $\lambda_{reldamp} = A p_{reldamp}$, where A is the contactor node equivalent area.

4.19.5 Frictional tractions and forces

4.19.5.1 Frictional constraint function, Cat 1 contact

The frictional Lagrange multipliers represent tangential forces. Consider a single contactor node. In 2D analysis, there is a single frictional Lagrange multiplier τ_t that represents the friction force in the t direction. In 3D analysis, there are two frictional Lagrange multipliers τ_t, τ_b that represent the friction force components in the t and b directions.

The frictional constraint functions are used to represent the relationship between slip velocity and tangential force. The terminology \dot{u} = magnitude of the slip velocity and τ_f = magnitude of the friction force is used.

Two friction algorithms are available.

Default algorithm: The default algorithm involves a more accurate linearization of the frictional constraints and, in general, converges much faster than its predecessor.

The frictional constraint functions enforce the following conditions:

$$0 = \tau_t + \frac{\dot{u}_t}{\dot{u}} \tau_f(\dot{u}, \tau_{fslide}) \quad (4.19-14a)$$

$$0 = \tau_b + \frac{\dot{u}_b}{\dot{u}} \tau_f(\dot{u}, \tau_{fslide}) \quad (4.19-14b)$$

$\tau_f(\dot{u}, \tau_{fslide})$ is the sliding friction force at the slip velocity with magnitude \dot{u} . This function is shown in Fig 4.19-6. Here ε_T is a small parameter (EPST parameter in the CGROUP commands) which has the physical meaning of the "sticking velocity", that is, the maximum velocity corresponding to sticking conditions.

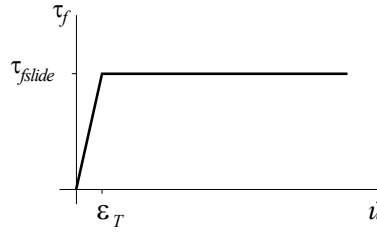


Figure 4.19-6: Frictional contact constraint function for default friction algorithm, Cat 1 contact

Version 8.3 algorithm: In the version 8.3 friction algorithm, the constraint function is defined implicitly via

$$|\tau_f| + v - \frac{2}{\pi} \arctan\left(\frac{\dot{u} - v}{\varepsilon_T}\right) = 0 \quad (4.19-15)$$

Here ε_T is a small parameter (EPST parameter in the CGROUP commands) which provides some elastic slip to the Coulomb friction law as shown in Fig. 4.19-7.

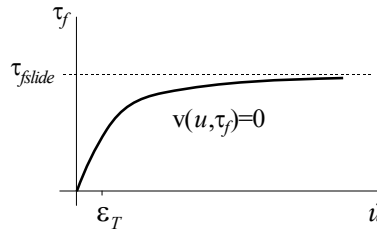


Figure 4.19-7: Frictional contact constraint function for version 8.3 friction algorithm, Cat 1 contact

The friction algorithm can be selected using the FRICTION-ALGORITHM parameter of the CONTACT-CONTROL command.

Note that the sliding force τ_{fslide} depends on the normal force λ_g and hence the constraint function has a dependence on the normal force. The dependence of the constraint function on the normal force is inherently non-symmetric, however this dependence is not included in the system matrices **KAV**, **KAW**, **KAA**.

4.19.5.2 Frictional constraint function, Cat 2 contact

The frictional Lagrange multipliers represent tangential tractions. Consider a single contactor node. In 2D analysis, there is a single frictional Lagrange multiplier T_t that represents the friction traction in the t direction. In 3D analysis, there are two frictional Lagrange multipliers T_t, T_b that represent the friction traction components in the t and b directions.

The frictional constraint functions enforce the following conditions:

$$0 = T_t + \frac{\dot{u}_t}{\dot{u}} T_f(\dot{u}, T_{fslide}) \quad (4.19-16a)$$

$$0 = T_b + \frac{\dot{u}_b}{\dot{u}} T_f(\dot{u}, T_{fslide}) \quad (4.19-16b)$$

$T_f(\dot{u}, T_{fslide})$ is the sliding friction traction at the slip velocity with magnitude \dot{u} . This function is shown in Fig 4.19-8. Here ε_T is a small parameter that has the physical meaning of the "sticking velocity", that is, the maximum velocity corresponding to sticking conditions.

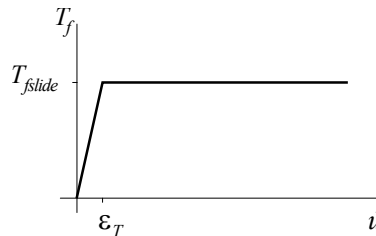


Figure 4.19-8: Dependence of sliding friction traction on slip velocity

Note that the sliding traction T_{fslide} depends on the gap pressure p_g and hence the frictional constraint functions have a dependence on the gap pressure. The dependence of the frictional constraint functions on the gap pressure is inherently non-symmetric. When a

symmetric solver is used, this dependence is not included in the system matrices \mathbf{KAV} , \mathbf{KAW} , \mathbf{KAA} , and when the nonsymmetric solver is used, this dependence is included in the system matrices \mathbf{KAV} , \mathbf{KAW} , \mathbf{KAA} .

4.19.6 Friction delay features

By default friction forces are generated immediately when contact is established, and the friction forces correspond to either sticking or slip conditions, as described above.

The friction delay features modify how the friction forces are applied. There are two options related to friction delay. Both of these options can be used in Cat 1 and Cat 2 contact.

Frictionless friction delay: Frictional conditions are applied to a contactor node one time step after contact is established (Fig 4.19-9). This feature can be useful in many problems, since it delays the non-linearity associated with friction until contact is established.

Stabilized friction delay: In the time step in which contact is established, stabilization is added to the frictional constraint function, if the relative sliding velocity corresponds to sliding conditions. This stabilization provides additional stiffness in the friction equations. If a solution is obtained, the frictional force is correctly computed (from the normal force and friction coefficient), however convergence might be slower than if stabilized friction delay is not used.

The friction delay feature is activated using the FRIC-DELAY parameter of the CGROUP commands, or by using the FRIC-DELAY parameter of the CONTACT-CONTROL command. Friction delay is off by default.

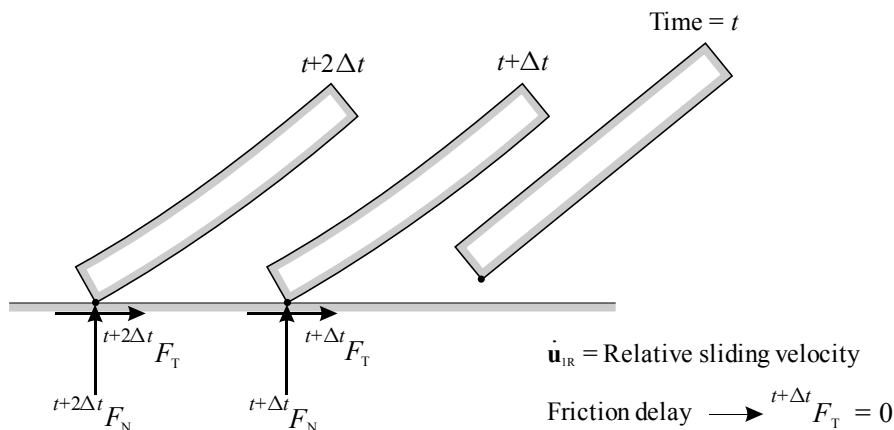


Figure 4.19-9: Frictionless friction delay feature

4.19.7 Consistent contact stiffness

Changes in the direction of the contact normal theoretically provide an additional contribution to the stiffness matrix that is proportional to the value of the contact force and the change in the normal direction. Therefore, higher convergence rates (closer to quadratic) can sometimes be obtained by selecting the consistent contact stiffness option, which accounts for these additional stiffness contributions.

However, consistent contact stiffness creates a direct connection between contactor and target nodes in the system matrices, as denoted by the Xs in the equation below.

$$\begin{bmatrix} \mathbf{K}_{UU} & \mathbf{K}_{UV} & \mathbf{K}_{UW} & \mathbf{0} \\ \mathbf{K}_{VU} & \mathbf{K}_{VV} & \mathbf{X} & \mathbf{K}_{V\Lambda} \\ \mathbf{K}_{WU} & \mathbf{X} & \mathbf{K}_{WW} & \mathbf{K}_{W\Lambda} \\ \mathbf{0} & \mathbf{K}_{\Lambda V} & \mathbf{K}_{\Lambda W} & \mathbf{K}_{\Lambda\Lambda} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{U} \\ \Delta \mathbf{V} \\ \Delta \mathbf{W} \\ \Delta \Lambda \end{bmatrix} = \begin{bmatrix} \mathbf{R}\mathbf{U} - \mathbf{F}\mathbf{U} \\ \mathbf{R}\mathbf{V} + \Lambda - \mathbf{F}\mathbf{V} \\ \mathbf{R}\mathbf{W} - \Lambda - \mathbf{F}\mathbf{W} \\ -\mathbf{w}(\mathbf{V}, \mathbf{W}, \Lambda) \end{bmatrix} \quad (4.19-17)$$

This direct connection leads to an increase in the number of non-zero entries in the stiffness matrix, and hence an increase in the memory required to store the stiffness matrix and an increase in the

solution time required to solve the system of equations. These increases are detrimental for large problems.

The consistent contact stiffness option is more beneficial when segment contact normals are selected, because the derivation assumes that the segment contact normals option is used.

The consistent contact stiffness feature is not used when the target surface is rigid.

Consistent contact stiffness is not used in dynamic analysis.

The consistent contact stiffness feature is set via the CONSISTENT-STIFF parameter of the CGROUP commands. The consistent contact stiffness feature is activated by default when all of the following conditions are satisfied: 1) segment normals are used; 2) the old contact surfaces of Cat 1 contact are used.

Consistent contact stiffness is never used for Cat 2 contact, and the setting of CONSISTENT-STIFF is ignored for Cat 2 contact.

4.19.8 Post-impact corrections

In post-impact corrections, the velocities and accelerations of the contactor and target can be forced to be compatible during contact (only in the normal contact direction). This is achieved by modifying the velocities and accelerations of the contact nodes once convergence is reached such that they satisfy conservation of linear and angular momentum.

The post-impact correction option requires additional memory and computations.

The post-impact correction feature should not be used together with compliant contact surfaces, since the velocities and accelerations of the contactor and target surfaces are no longer expected to be identical.

If post-impact correction is activated, all target nodes, except those with all degrees of freedom fixed or enforced displacements, must have a positive non-zero mass. The contactor nodes can have zero

mass.

Setting the Newmark $\alpha = 0.5$ instead of the default $\alpha = 0.25$ (trapezoidal rule — see Section 7.3) results in an accurate solution of rigid body impact problems, and frequently has a positive effect on reducing numerical oscillations in flexible body contact. Therefore the post-impact correction feature includes an option to set $\alpha = 0.5$.

It is, however, recommended that the Bathe method be used instead, whenever possible.

Post-impact corrections are activated using the POSTIMPACT parameter of the CONTACT-CONTROL command. The following options are available:

DEFAULT - either NO or YES, see below.

YES - perform post-impact corrections. This is the default when using the kinematic constraint method with the Noh-Bathe method.

NO - do not perform post-impact corrections.

MOD - set the Newmark time integration parameter ALPHA to 0.5.

Post-impact corrections are not considered a best practice, and are not supported in Cat 2 contact.

4.19.9 Heat transfer between contact surfaces

Section 4.6.1 discusses the theory of heat transfer between contact surfaces used in thermo-mechanical coupling. From that section, the term

$$\sum_{\text{Contactor nodes}} q^{SC} A_c (\delta\theta^C - \delta\theta^T) \quad (4.19-18)$$

where $q^{SC} = \omega \hat{h} (\theta^T - \theta^C)$, is added to the principle of virtual temperatures to account for heat transfer between contact surfaces.

Therefore for a single contactor node k with equivalent area A_k and temperature θ^k , the term

$$\omega \hat{h}(\theta^T - \theta^k) A_k (\delta\theta^k - \delta\theta^T) \quad (4.19-19)$$

is added to the principle of virtual temperatures. It is necessary to express the temperature θ^T on the target segment in terms of the temperatures θ^{kT} of the target segment nodes. This is done using

$$\theta^T = \sum_{kT} h_{kT} \theta^{kT} \quad (4.19-20)$$

in which h_{kT} are the target node shape functions. The addition to the right-hand-side of the equations of thermal equilibrium (nodal heat flux load vector) is

$$-\omega \hat{h} A_k (\theta^k - \theta^T) \begin{bmatrix} 1 \\ -h_1 \\ -h_2 \\ \vdots \end{bmatrix} \quad (4.19-21)$$

and the addition to the left-hand-side of the equations of thermal equilibrium (thermal conductivity matrix) is

$$\omega \hat{h} A_k \begin{bmatrix} 1 & -h_1 & -h_2 & \dots \\ & h_1 h_1 & h_1 h_2 & \dots \\ sym & & h_2 h_2 & \dots \\ & & & \ddots \end{bmatrix} \quad (4.19-22)$$

Equation (4.19-21) shows that the presence of contact heat transfer creates direct connections between contactor and target thermal degrees of freedom. These direct connections change as the state of contact changes.

4.19.10 Heat generation due to friction

Section 4.6.2 discusses the theory of heat generation due to friction thermo-mechanical coupling. From that section, the term

$$\sum_{\text{Contactor nodes}} \dot{e}_f^S A_c (f_C \delta\theta^C + f_T \delta\theta^T) \quad (4.19-23)$$

where $\dot{e}_f^S A_c$ is the rate of frictional energy dissipation at the contactor node, is added to the principle of virtual temperatures to account for heat generation due to friction.

Therefore for a single contactor node k with equivalent area A_k and temperature θ^k , the term

$$\dot{e}_f^S A_k (f_C \delta\theta^k + f_T \delta\theta^T) \quad (4.19-24)$$

is added to the principle of virtual temperatures. Again using the target segment shape functions, the addition to the right-hand-side of the equations of thermal equilibrium (nodal heat flux load vector) is

$$\dot{e}_f^S A_k \begin{bmatrix} f_C \\ f_T h_1 \\ f_T h_2 \\ \vdots \end{bmatrix} \quad (4.19-25)$$

There is no addition to the left-hand-side of the equations of thermal equilibrium.

4.19.11 Heat transfer and heat generation with the option of using corner nodes in thermal calculations

It is allowed to only use corner nodes in the thermal calculations, by selecting TMC-CONTROL TEMP-INTERPOLATION=CORNER.

This option is supported in Cat 1 contact only in 3D contact when 9-node target segments are used.

This option is supported in Cat 2 contact for both 2D and 3D contact, for all higher-order target segments.

When the option of using corner nodes in the thermal solution is selected then, in the thermal calculations, higher-order target segments are replaced by lower-order target segments, so that only the corner target segment nodes participate in the thermal calculations.

For heat generation due to friction, the frictional energy dissipated at mid-side contactor nodes is lumped onto the corner contactor nodes, in such a way that the total frictional energy dissipation is unchanged.

4.20 Contactor elements in explicit analysis

For Cat 1 contact, in explicit analysis, either the penalty method or the kinematic constraint method is used. The penalty method is the default when the Noh-Bathe method is used. The kinematic constraint method is the default when the central difference method is used.

The penalty method is the simplest and fastest of the explicit contact algorithms. It can also handle rigid contactor and target surfaces.

The main disadvantage of the penalty method is that contact conditions are not exactly satisfied and it usually shows oscillations in contact forces. These oscillations can sometimes be removed by using the penalty rate damping factor K_D . The penalty method is

also sensitive to the choice of the penalty stiffness K_N . If that stiffness is too large it leads to instability and oscillations, and if it is too small it leads to excessive penetrations.

The default penalty stiffness selected by the program is, in most cases, a suitable compromise.

For Cat 2 contact, in explicit analysis, only the penalty method is available.

4.21 Contactor element forces in explicit analysis with the penalty method

In explicit analysis, there are no Lagrange multipliers or constraint functions.

4.21.1 Normal forces

In Cat 1 contact, when an overlap is detected, the contact pressure at the contactor node is computed using equation (4.2-26) in Section 4.2.3.2.

If the Cat 2 CNORMAL commands are used to specify the normal contact response, then the contact pressure is computed using the contact pressure functions given in Section 4.2.2.

4.21.2 Relative contact damping forces

In Cat 1 contact, the relative contact damping pressure is computed using equation (4.5-7) of Section 4.5.3.

In Cat 2 contact, the relative contact damping pressure is computed using either equation (4.5-7) of Section 4.5.3 or using the formulas of Section 4.5.2.

The contact damping force is $\lambda_{reldamp} = Ap_{reldamp}$ where A is the contactor node equivalent area.

4.21.3 Frictional tractions and forces

For Cat 1 contact the frictional force is computed as

$$\mathbf{F}_T = -\min\left(A K_T \mathbf{x}_T, \mu(\lambda_g + \lambda_{reldamp}) \frac{\mathbf{x}_T}{\|\mathbf{x}_T\|}\right) \quad (4.21-1)$$

where \mathbf{x}_T is the relative tangential incremental displacement and K_T is the tangential penalty damping. Only Coulomb friction can be used in Cat 1 contact. Notice that the relative damping force appears in the above equation.

The tangential penalty damping K_T can be selected by the user, or determined automatically by the program.

For Cat 2 contact, the magnitude of the frictional traction is computed as

$$T_f = \min(T_{fslide}, T_{fstick}) \quad (4.21-2)$$

in which

$$T_{fstick} = T_{fslide} \frac{\dot{u}}{\epsilon_T}, \quad \text{EPST used} \quad (4.21-3)$$

or

$$T_{fstick} = K_T \dot{u}, \quad \text{tangential penalty damping used} \quad (4.21-4)$$

T_{fslide} is based solely on the gap pressure p_g and does not include any effects from relative contact damping, see Section 4.3 for the calculation of T_{fslide} .

Notice that in Cat 2 contact, K_T has a different meaning than in Cat 1 contact. Roughly speaking,

$$K_T|_{\text{Cat } 2} = \frac{K_T|_{\text{Cat } 1}}{\Delta t} \quad (4.21-5)$$

Therefore the automatic calculation of K_T is different in Cat 2 contact than in Cat 1 contact.

4.22 Contactor element forces in explicit analysis with the kinematic constraint method

4.22.1 Normal forces

The normal contactor element forces in explicit analysis with the kinematic constraint method are described in Section 4.2.3.1.

4.22.2 Frictional forces

The frictional contactor element forces in explicit analysis with the kinematic constraint method are described in Section 4.3.7.1.

4.23 Contact output

4.23.1 Contact tractions and status variables

Contact tractions and status variables are output when TRACTIONS=YES in the CGROUP command.

4.23.1.1 Status variable - contactor node states

The possible states of the contactor nodes are

Open, 2: no target segment is found for the contactor node, or the gap between the contactor node and target segment is open.

Closed, 3: the gap between the contactor node and the target segment is closed; a compressive force is acting onto the contactor node and frictionless conditions are assumed.

Sticking, 4: the gap between the contactor node and target segment is closed; a compressive force is acting onto the contactor node and the relative tangential velocity is less than the sticking velocity. The frictional traction is less than the sliding frictional traction.

Sliding, 5: the gap between the contactor node and the target segment is closed; a compressive force is acting onto the contactor node and the relative tangential velocity is greater than the sticking velocity. The frictional traction is equal to the sliding frictional traction.

Reldamp, 6: the gap between the contactor node and the target segment is open; only relative contact damping forces act onto the contactor node.

The state of a contactor node can be accessed in the AUI using the AUI variable `NODAL_CONTACT_STATUS`.

4.23.1.2 Gap and overlap variables

The gap and overlap of a contactor node can be accessed in the AUI using the variables `NODAL_CONTACT_GAP` and `NODAL_CONTACT_OVERLAP`.

When relative contact damping is requested, the gap velocity and overlap velocity can be accessed in the AUI using the variables `NODAL_CONTACT_GAP_VELOCITY` and `NODAL_CONTACT_OVERLAP_VELOCITY`.

4.23.1.3 Slip velocity variables

When friction is included, the relative tangential velocities (slip velocities) can be accessed in the AUI using the variables

```
NODAL_SLIP_VELOCITY
                                (magnitude of slip velocity)
NODAL_SLIP_VELOCITY-{XYZ}
```

The magnitude of the slip velocity of contactor nodes that are currently sticking can be accessed in the AUI using the variable

NODAL_STICK_VELOCITY

This variable is set to 0 if the node is currently sliding.

The nodal sliding distance can be accessed in the AUI using the variable NODAL_SLIDING_DISTANCE.

4.23.1.4 Segment-based contact tractions

In Cat 1 contact, when the old contact surface representation is used, the tractions are segment-based (see Section 4.8.1.9 for a description of the old contact surface representation).

Traction output is available only on contactor surfaces, and only if the surface is segment-based (not nodal based). Each contactor contact segment outputs, at its center, the following output information to the porthole file. This information is accessible in the AUI using the given variable names.

2-D contact: SEGMENT_LENGTH, NORMAL-{YZ},
NORMAL_TRACTION, TANGENTIAL_TRACTION

3-D contact: SEGMENT_AREA,
NORMAL-{XYZ}, NORMAL_TRACTION-{XYZ},
TANGENTIAL_TRACTION-{XYZ},

In addition, the following information is output for solitary nodes: (solitary nodes are nodes having concentrated contact forces not resulting from segment tractions, e.g., when an isolated node is in contact):

SOLITARY_CONTACT_FORCE-X (3-D contact only),
SOLITARY_CONTACT_FORCE-Y,
SOLITARY_CONTACT_FORCE-Z,
SOLITARY_CONTACT_FORCE-A (3-D contact only),
SOLITARY_CONTACT_FORCE-B,
SOLITARY_CONTACT_FORCE-C.

It should be noted that if symmetric contact pairs are used, then the tractions output are one-half of the expected values. This is because the tractions are computed solely from the contactor forces, and when symmetric contact pairs are used, the contactor

forces are one-half of the total forces.

4.23.1.5 Nodal-based contact tractions

In Cat 1 contact, when the new contact surface representation is used, or in Cat 2 contact, the tractions are nodal-based.

Output is restricted to the contactor nodes, and only if the surface is segment-based (not nodal based). The output data is available to the AUI using the variable names given below.

```
NODAL_NORMAL_TRACTION
      (magnitude of normal traction)
NODAL_NORMAL_TRACTION-{XYZ}
NODAL_TANGENTIAL_TRACTION
      (magnitude of tangential traction)
NODAL_TANGENTIAL_TRACTION-{XYZ}
```

All of these variables are output at the nodes. As a convenience, band plots also allow the variable names

```
NORMAL_TRACTION
NORMAL_TRACTION-{XYZ}
TANGENTIAL_TRACTION
TANGENTIAL_TRACTION-{XYZ}
```

When relative contact damping is used, then the above normal traction variables refer to the gap pressures. The following variables can be used to examine the relative contact damping pressures:

```
NODAL_RELDAMP_TRACTION
      (magnitude of damping traction)
NODAL_RELDAMP_TRACTION-{XYZ}
```

Nodal-based traction options

In Cat 1 contact, there are two options for nodal-based tractions. The current traction calculation is selected using CONTACT-CONTROL ... TRACT-CALC=CURRENT (the default). In Cat 1 contact, it is also possible to use the traction calculation of versions 8.9 and lower, and the 8.9 traction calculation is selected using

CONTACT-CONTROL ... TRACT-CALC=V89.

In Cat 2 contact, only the current traction calculation is allowed.

Symmetric contact and self-contact

When the current traction calculation is used, then a special procedure is used for symmetric contact and self-contact, as follows:

If symmetric contact is used, the contactor node tractions are computed from both the contactor contact forces and also the target contact forces (projected to the contactor nodes).

To be specific, suppose that contact surfaces A and B are used in symmetric contact, such that in pair 1, A is the contactor and B is the target, and in pair 2, A is the target and B is the contactor. In the calculation of contact tractions on surface A, the contactor node forces from A are used (since A is a contactor in pair 1) and also the contactor node forces from B are used, projected onto A (since A is a target in pair 2). Similarly, in the calculation of contact tractions on surface B, the contactor node forces from B are used, and also the contactor node forces from A are used, projected onto B.

This procedure gives reasonable contact tractions for both surfaces A and B.

Self-contact is considered to be a special case of symmetric contact, and the same projection scheme is used in self-contact.

When the version 8.9 traction algorithm is used in conjunction with symmetric contact and self-contact, then the tractions output are one-half of the expected values. This is because the tractions are computed solely from the contactor forces, and when symmetric contact pairs are used, the contactor forces are one-half of the total forces.

Contact geometry and tractions

When the current traction calculation is used, the geometry of the contact affects the calculations as shown in Fig. 4.23-1. Consider the calculation of the traction at node B. The topologies of Figs. 4.23-1(a) and 4.23-1(b) are identical, but in 4.23-1(a), segment B-C contributes to the traction calculation at node B, and in 4.23-1(b), segment B-C does not contribute to the traction calculation at node B. The criterion TRACT-ANGLE can be set by CONTACT-CONTROL ... TRACT-ANGLE; the default is 45 degrees.

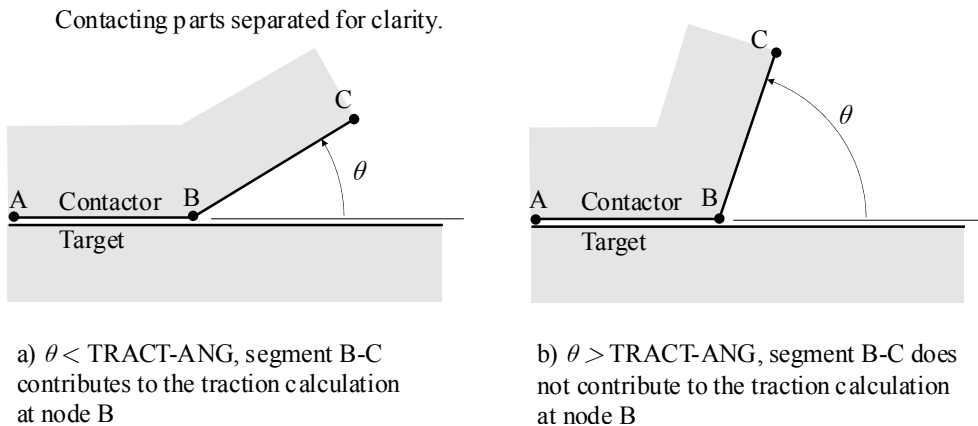


Figure 4.23-1: Explanation of TRACT-ANGLE parameter

Tractions at corner nodes of higher-order contact segments

The tractions at the corner nodes of higher-order segments are computed by interpolating the tractions from the adjacent mid-side and mid-face nodes. It is possible to change this interpolation using CONTACT-CONTROL ... TRACT-QUADRATIC. Allowed values of TRACT-QUADRATIC are: EDGEAVERAGE, averages values from two adjacent nodes; AVERAGE, averages values from all adjacent nodes; MAXIMUM, uses maximum value of any adjacent node. EDGEAVERAGE is the default.

4.23.2 Contact forces

Contact forces are output when FORCES=YES in the CGROUP commands. These forces are sometimes called "consistent contact forces".

Force output is available for all contact nodes under all contact conditions. The AUI variables are:

```
CONTACT_FORCE- { XYZ }
CONTACT_FORCE- { ABC }
```

When relative contact damping is used, then the above contact forces contain normal forces from gap pressures only. The following variables can be used to access the forces due to relative contact damping:

```
CONTACT_RELDAMP_FORCE- { XYZ }
```

4.24 Contact convergence in implicit analysis

The system of equations to be solved has the form given in equation (4.19-5), which we repeat here:

$$\begin{bmatrix} \mathbf{KUU} & \mathbf{KUV} & \mathbf{KUW} & \mathbf{0} \\ \mathbf{KVU} & \mathbf{KVV} & \mathbf{0} & \mathbf{KVA} \\ \mathbf{KWU} & \mathbf{0} & \mathbf{KWW} & \mathbf{KWA} \\ \mathbf{0} & \mathbf{KAV} & \mathbf{KAW} & \mathbf{KAA} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{U} \\ \Delta \mathbf{V} \\ \Delta \mathbf{W} \\ \Delta \mathbf{A} \end{bmatrix} = \begin{bmatrix} \mathbf{RU} - \mathbf{FU} \\ \mathbf{RV} + \mathbf{FA} - \mathbf{FV} \\ \mathbf{RW} + \mathbf{HA} - \mathbf{FW} \\ -\mathbf{w}(\mathbf{V}, \mathbf{W}, \mathbf{A}) \end{bmatrix} \quad (4.24-1)$$

When the right-hand-side of the system of equations is small, the solution has converged.

Due to the Lagrange multiplier degrees of freedom used in contact, additional contact related norms are used to assess convergence. The contact related norms are given in the intermediate printout of convergence information in the ADINA output listing.

Box f of Fig. 4.24-1 shows the contact related norms. Parameter CFORCE indicates the norm of the change in the contact forces (between two iterations).

$$\text{CFORCE} = \|\mathbf{F}\mathbf{A}^{(i)} - \mathbf{F}\mathbf{A}^{(i-1)}\| \quad (4.24-2)$$

Parameter CFNORM gives the norm of the contact force vector

$$\text{CFNORM} = \|\mathbf{F}\mathbf{A}^{(i)}\| \quad (4.24-3)$$

Box f also shows the node and value for the max entry in CFORCE. This information might be useful for determining the causes of difficulties in convergence.

The following additional convergence criterion is used:

$$\frac{\text{CFORCE}}{\max(\text{CFNORM}, \text{RCONSM})} \leq \text{RCTOL} \quad (4.24-4)$$

where RCTOL is the contact force convergence tolerance and RCONSM is the reference contact force to prevent possible division by zero in the contact convergence criterion above. By default, RCTOL=0.05 and RCONSM=0.01.

When the maximum number of iterations is reached without convergence, and all norms are decreasing, the maximum number of iterations should be increased.

OUT-OF-BALANCE ENERGY		NORM OF				
		OUT-OF-BALANCE		INCREMENTAL		
		FORCE	MOMENT	DISP.	ROTN.	CFORCE
		NODE-DOF MAX VALUE	NODE-DOF MAX VALUE	NODE-DOF MAX VALUE	NODE-DOF MAX VALUE	NODE MAX VALUE CFNORM
		box b	box c	box d	box e	box f
ITE= 0	2.77E+01	4.26E+01	0.00E+00	2.75E+00	0.00E+00	2.77E-08
	459	0	2313	0	1912	
	4.48E+00	0.00E+00	-1.06E-01	0.00E+00	1.97E-08	3.84E+00
ITE= 1	4.07E-03	4.07E+01	0.00E+00	6.23E-03	0.00E+00	3.90E+00
	1014-X	0	2011-Z	0	2414	
	-4.77E+00	0.00E+00	2.07E-04	0.00E+00	2.64E+00	7.03E+00
ITE= 2	8.06E-12	9.51E-05	0.00E+00	4.53E-06	0.00E+00	2.20E+00
	1067-Z	0	69-X	0	2414	
	-1.13E-05	0.00E+00	9.35E-08	0.00E+00	-1.49E+00	5.15E+00
ITE= 3	1.67E-20	2.96E-07	0.00E+00	2.82E-10	0.00E+00	2.61E-03
	2417-X	0	1911-X	0	1911	
	-5.84E-08	0.00E+00	7.30E-12	0.00E+00	-1.77E-03	5.15E+00

```

... CONVERGENCE RATIOS   CONVERGENCE RATIOS   OUT-OF-BALANCE LOAD
... FOR OUT-OF-BALANCE   FOR INCREMENTAL   VECTOR CALCULATION
...   ENERGY             FORCE             DISP.             CFORCE             BETA             RATIO
...                       MOMENT           ROTN.              (ITERNS)

...   COMPARE WITH         COMPARE WITH
...   ETOL                RTOL                DTOL                RCTOL
...   1.00E-03            1.00E-02   (NOT USED)   1.00E-03

...   1.00E+00            1.00E+00   1.66E+00   7.21E-09
...   0.00E+00            0.00E+00

...   1.47E-04            9.55E-01   3.76E-03   5.54E-01   1.00E+00   -2.09E-04
...   0.00E+00            0.00E+00   ( 1)

...   2.91E-13            2.23E-06   2.73E-06   4.27E-01   1.00E+00   4.71E-06
...   0.00E+00            0.00E+00   ( 1)

...   6.03E-22            6.94E-09   1.70E-10   5.07E-04   1.00E+00   0.00E+00
...   0.00E+00            0.00E+00   ( 1)

```

Figure 4.24-1: ADINA output listing of convergence criteria during equilibrium iterations

When the norms are rapidly changing before convergence fails, it is commonly caused by applying the load too quickly or using a large time step.

When CFNORM is stable but CFORCE changes rapidly during equilibrium iterations, the contact can be oscillating between two or more close solutions. In this case, try to change the time stepping, or turn on the suppression of contact oscillations feature.

Premature convergence in the out-of-balance force norm

In many analyses, the out-of-balance force norm is used to assess convergence. However, the out-of-balance force norm, if used by itself, can give misleading results in contact analysis. Consider a single degree of freedom system with a linear spring element in contact. In Cat 1 contact, the system of equations is

$$\begin{bmatrix} k & -1 \\ * & * \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta \lambda_g \end{bmatrix} = \begin{bmatrix} R + \lambda_g - ku \\ w(g, \lambda_g) \end{bmatrix} \quad (4.24-5)$$

with out-of-balance force norm equal to $|R + \lambda_g - ku|$. After the solution for $\Delta u, \Delta \lambda_g$, the out-of-balance force norm is equal to zero, no matter what the constraint function value is, because the non-contact part of the model is linear. Hence just because the out-of-balance force norm is small, the solution might not be converged.

In Cat 2 contact, the system of equations is

$$\begin{bmatrix} k & -A \\ * & * \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta p_g \end{bmatrix} = \begin{bmatrix} R + Ap_g - ku \\ w(g, p_g) \end{bmatrix} \quad (4.24-6)$$

with out-of-balance force norm equal to $|R + Ap_g - ku|$. After the solution for $\Delta u, \Delta p_g$, the out-of-balance force norm might not equal zero, because the equivalent area A might have changed.

In general, due to the equivalent areas appearing on the right-hand-side of the system of equations in Cat 2 contact, the out-of-balance force vector in Cat 2 contact tends to converge more slowly to zero than the out-of-balance force vector in Cat 1 contact.

In both Cat 1 and Cat 2 contact, it is important to use an additional criterion to assess convergence, for example, the contact convergence criterion.

Contact and line searches

Because line searches try to minimize the out-of-balance force vector, without consideration of the contact constraint equations, it is possible that the use of line searches can cause the equilibrium iterations to converge more slowly. The actual performance of line searches is very problem-dependent.

Solution diagnostics

The contact solution diagnostics, printed in the .out file when SOLUTION=DETAILED in the DIAGNOSTICS command, are very useful in determining whether a solution has converged. The following diagnostics are particularly useful:

```
Nodes coming into contact ... nodes including
node ... with penetration = ...
```

```
Maximum penetration ... ( ... % of model size) at
node ....
```

```
Nodes losing contact . ... nodes including node
... with contact force ...
```

```
Nodes losing contact . ... nodes including node
... with previous penetration ...
```

```
Nodes losing contact by exceeding boundaries of
target surfaces ... nodes including node ...
```

```
All contact is lost
```

```
Nodes changing target segment ... nodes including
node ... with change of penetration ...
```

```
Nodes switching from stick to slip ... nodes  
including node .. with contact force increase  
..., slip velocity=...
```

```
Nodes switching from slip to stick ... nodes  
including node .. with contact force...
```

Recommendations

Generally, if the nodes are entering/exiting contact, or changing stick/slip status, the solution has not converged yet, and if the equilibrium iterations have finished, the convergence tolerances were too loose.

Pay attention to the penetrations reported by the diagnostics. If the penetrations reported by the diagnostics are unphysically large, then it is quite possible that there is a modeling error. For example, if a node changes target segment with a large change of penetration, it is possible that the contact search has found the wrong target segment.

Nodes should not lose contact due to exceeding boundaries of the target surfaces. If this diagnostic appears in the output listing, check the modeling. The contact surface extension factor might need to be increased, or the sizes of the target surfaces might need to be increased.

Solution graph feature

The solution graph feature can be used to create graphs of contact output, for example, the total number of nodes in and out of contact for each time step.

4.25 Frequency analysis

It is allowed to perform frequency analysis when contact surfaces are in the model.

In Cat 1 contact, the frequency analysis need not be performed after a static or dynamic analysis. During the frequency analysis run, the program first updates the contact conditions, then performs the

frequency analysis. If the contact constants (such as the coefficient of friction) are changed in the frequency analysis, the updated contact constants are used in the frequency analysis.

In Cat 2 contact, the frequency analysis must be performed after a static or dynamic analysis. The state of contact obtained in the preceding analysis is used to determine contact conditions during the frequency analysis. If the contact constants are changed in the frequency analysis, the updated contact constants are not used in the frequency analysis.

The determinant search method cannot be used for frequency analysis with contact.

4.25.1 Normal contact

A contactor node that is in contact with a target segment remains connected to the target segment in the frequency analysis. This connection takes the form of a spring connecting the contactor node and target segment in the normal direction.

There are two options:

FREQUENCIES CONT-COMPL=NO

The connection between the contactor node and target segment is assumed to be extremely stiff, no matter what the contact force is just before the frequency analysis, and no matter what law is used for the normal contact response.

FREQUENCIES CONT-COMPL=YES (the default)

The connection between the contactor node and target segment is based on the contact force just before the frequency analysis and the law used for the normal contact response. For example, if compliant contact is used for the normal contact response, the spring connecting the contactor node and target segment has stiffness based on the contact compliance.

4.25.2 Frictional contact

When frictional contact is used, a contactor node that is in contact with a target segment might or might not remain connected in the tangential directions, depending on the state of friction. This connection takes the form of springs (1 spring in 2D and 2 springs in 3D) connecting the contactor node and target segment in the tangential directions.

The algorithms are slightly different in Cat 1 and Cat 2 contact.

Cat 1 contact

If the contactor node is sticking just before the frequency solution, then the tangential connection is very stiff.

If the contactor node is sliding just before the frequency solution, then there is no tangential connection.

Cat 2 contact

If the contactor node is sticking at the end of the preceding analysis, then the tangential connection is very stiff.

If the contactor node is sliding at the end of the preceding analysis then there is no tangential connection.

All cases

The stiffness of the tangential connection does not depend on the magnitude of the normal contact force.

There is no way to control the stiffness of the tangential connections.

The nonsymmetric terms used in Cat 2 contact are not included in frequency analysis.

4.26 General modeling recommendations

4.26.1 Choice of contactor and target

- For some contact problems, the contactor and target surfaces in a contact pair can be interchanged without much effect on the solution. However, for many cases, one of the two alternatives is better.
- If it is more important for the nodes of one surface not to penetrate the other, then that surface should be the contactor. This factor is usually important when one surface has a much coarser mesh than the other as shown in Fig. 4.26-1. The coarse surface should preferably be the target in this case. A related condition occurs around corners or edges as shown in Fig. 4.26-2. The upper surface should preferably be the contactor in this case.

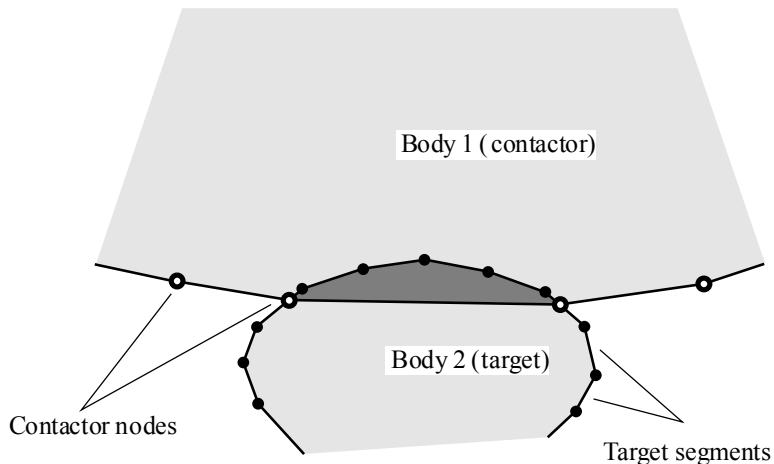


Figure 4.26-1: Effect of incorrect contactor-target selection due to mesh density

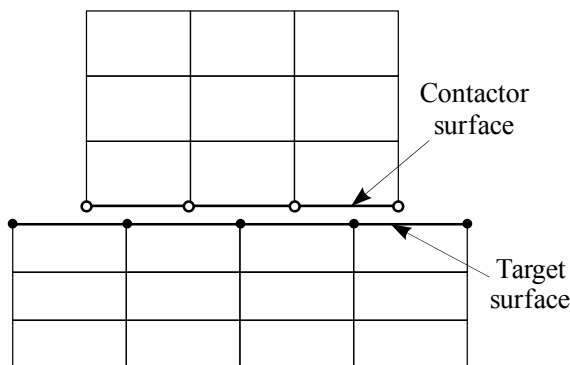


Figure 4.26-2: Target selection for surfaces of different sizes

- If one of the surfaces has mostly dependent degrees of freedom, it should be the target. This dependency can be due to boundary conditions, constraints or rigid elements.
- If the nodes of the surface are not attached to any elements, then the surface should be the target. However if compliant contact or the explicit penalty algorithm are used, then it is permitted for the surface to be the contactor.
- If one surface is significantly stiffer than the other, it should preferably be the target, unless one of the two conditions above also exist.

4.26.2 Benefits of compliant contact, compared with no-overlap contact

The benefits of contact compliance, compared with no-overlap contact, are that contact compliance:

Improves the convergence rate. By adding compliance to the contact surface, more contactor nodes come into contact with the target surface. It is better for convergence to have many nodes in contact with small forces at each contactor node, than to have few nodes in contact with large forces at each contactor node.

Improves the contact tractions. Without contact compliance, the discretization error of the mesh often cause some contactor

nodes on the contact area to not be in contact which results in spotty contact (or patchy) contact tractions. An example of spotty contact is shown in Fig. 4.26-3.

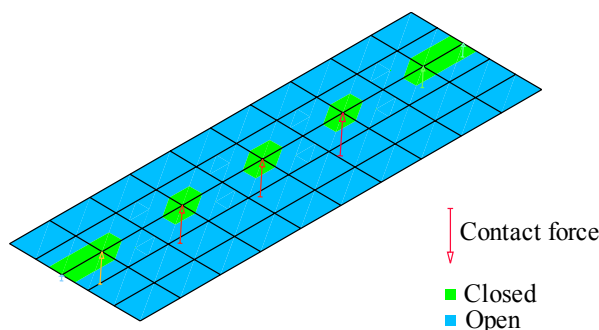


Figure 4.26-3: Spotty contact on a plate

Reduces numerical oscillations that sometimes result from dynamic time integration. Numerical oscillations, such as chatter or ringing, can be triggered by contact reversal. A node that is out of contact at time $t - \Delta t$, in contact at time t and out of contact at time $t + \Delta t$ is said to have had contact reversal. Contact surface compliance can be used to prevent these oscillations. For example, consider Fig. 4.26-4, where verification problem B.120 is solved using the Bathe method rather than the central difference method. The results show that no numerical oscillations are triggered when contact compliance is used.

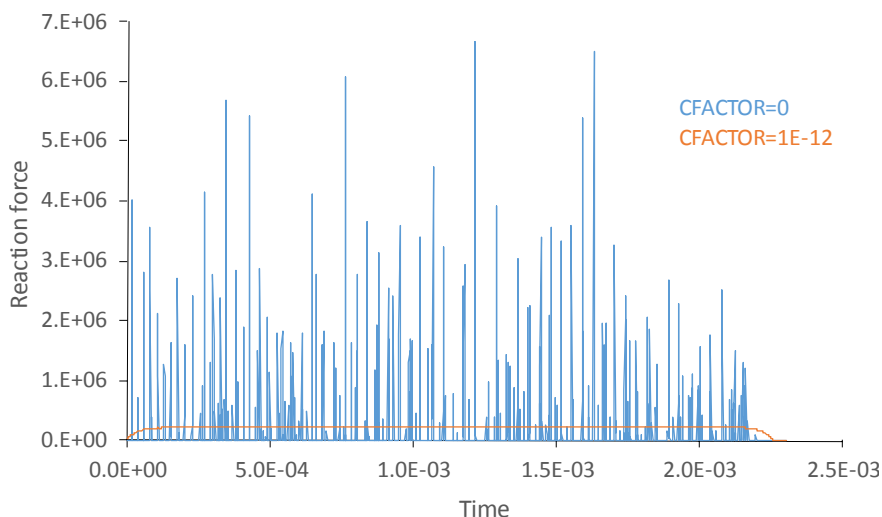


Figure 4.26-4: Reaction force from a thin rod impacting a rigid surface solved using the Bathe method (verification problem B.120).

However, the drawbacks of contact compliance are:

Nodal contact overlap. The contactor nodes penetrate the target segment when contact compliance is used. The compliance factor must be selected such that the contact pressures do not cause excessive nodal overlap.

Additional (spurious) energy is stored in the contact pair. Elastic energy is stored in the compliant contact surface. This energy is released when the contact pair opens which can have an effect in certain dynamic problems.

Can reduce the resultant contact force and stresses. If a prescribed displacement is applied, the contact surface compliance feature will reduce the stress in the body for a given displacement. The stresses are not reduced if the body is subjected to an applied load.

By default, the compliance factor is set to zero. However, in general, some amount of contact compliance should always be used. It is not default because the appropriate compliance factor is

model dependent. Since contact compliance is recommended, the following AUI Warning is printed when contact compliance is not used:

*** Warning: Contact surface compliance not used for contact group ? .
Contact compliance can improve the solution accuracy and convergence rate.

The value of the contact compliance factor is usually set by trial and error. A good starting value is 1×10^{-5} . The objective is to find the smallest compliance factor that results in convergence with good tractions (i.e., not spotty contact), but does not cause excessive penetration (nodal contact overlap).

After contact compliance is used, the amount of penetration should always be checked to make sure the maximum overlap is acceptable. The post-processing variable `NODAL_CONTACT_OVERLAP` can be used for this purpose.

When deciding on what penetration is acceptably small we must consider the mesh used. For example, if the target surface is curved, there will be a geometric error associated with using a coarse contractor surface. There is no advantage if the maximum penetration is less than the geometric error. So, if the target surface is coarse, a large maximum penetration can be used.

4.26.3 General modeling hints

- ADINA automatically determines the direction of the contact surfaces on the faces of solid elements. For contact surfaces defined on structural elements (beams, trusses and shells) the user has to ensure that the correct direction is defined (except in double-sided contact).
- Rigid target surfaces can be modeled using nodes with no degrees of freedom or nodes with prescribed displacements for all active degrees of freedom. As a result, a fine discretization of a complex rigid surface geometry is possible with only a small increase in the solution cost.
- If the contact surface is rigid it can be a slave to a master node that will control the motion of the rigid surface. Rigid links can be

conveniently used between the master node and all the nodes on the rigid surface.

- It is acceptable for the nodes on the contactor and target surfaces to be coincident (have identical coordinates). In this case, it is important to ensure that the two surfaces do not share the same nodes.
- If required, a contactor surface can be modeled as almost rigid by choosing a reasonably high Young's modulus for the finite elements modeling the contactor surface. However, the stiffness of the surface elements should not be excessively high and make the model ill-conditioned.
- If the degrees of freedom of a node on a contactor surface are used in constraint equations or attached to a rigid element (see Section 5.15.2), the contactor node degrees of freedom should preferably be independent.
- If the contact surfaces are smooth (i.e., the coefficient of friction is small), the frictionless model is recommended as it is less costly to use. It is also recommended that prior to any contact analysis involving friction, the frictionless solution is first obtained, whenever possible.
- A contactor node should preferably not belong to more than one contact surface, otherwise the contactor node may be over-constrained. If it is necessary for a contactor node to belong to more than one contact surface, then contact compliance should be used.
- For problems in which the contactor and target surfaces are initially relatively close to each other and no significant sliding between these surfaces is expected throughout the analysis, the small displacement contact feature may be used. The analysis will be faster in this case, since the relatively time consuming contact search is only performed once, and convergence difficulties associated with a contactor node oscillating between one target segment to another are eliminated. It is the user's responsibility however, to make sure that the problem is suitable for small displacement contact.

- Restarting from frictionless contact to contact with friction and vice versa is not possible. However, it can be done if the frictionless analysis is replaced by a frictional analysis with a very small friction coefficient.
- If a contact surface with corners or edges is modeled with continuous contact normals, the normal vectors may be inaccurate as shown in Fig. 4.26-5(a). In this case, switch to segment normals, use different contact surfaces for each smooth part (Fig. 4.26-5(b)) or use a fine mesh close to the corners or edges (Fig. 4.26-5(c)).

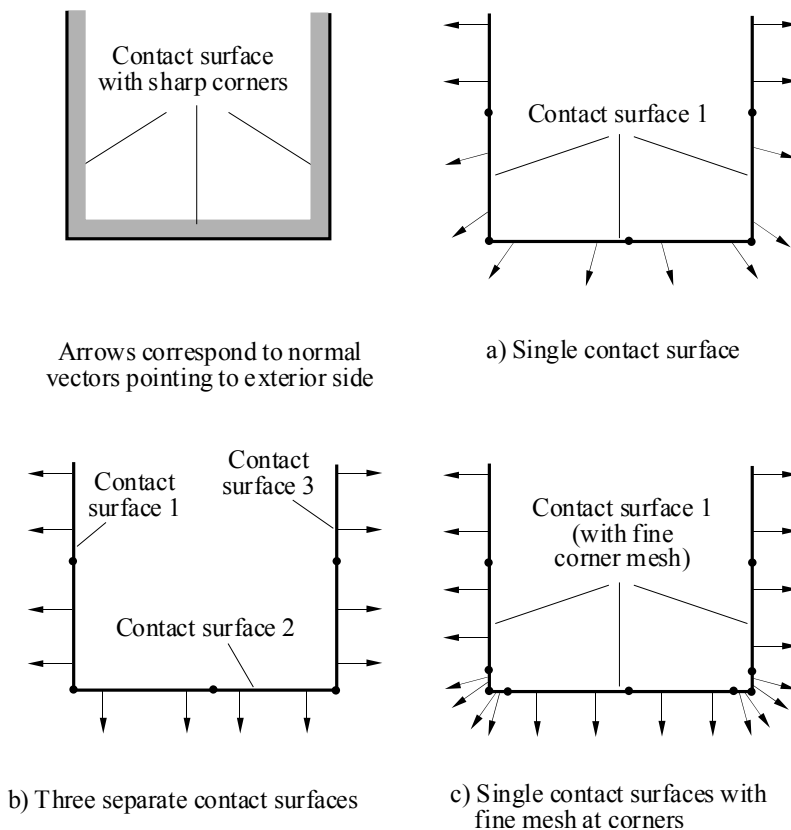


Figure 4.26-5: Defining contact surfaces (with continuous normal vectors) in the presence of corners

- In Cat 2 contact, the SEARCHDIST feature can be used to speed up the contact search, and to prevent incorrect contact searches when the target surfaces are open, see Section 4.11.2 for details and examples.

4.27 Modeling recommendations in implicit analysis

- It is recommended that the ATS method be used in contact analysis (see Section 7.2.1). It can also be effective to use the low-speed dynamics option of the ATS method (AUTOMATIC TIME-STEPPING RESPS=YES).
- Line search can sometimes be beneficial for contact problems.
- Contact compliance has many advantages and should be used whenever possible, as discussed in detail in Section 4.26.2.
- For problems that have difficulties in convergence because contactor nodes are changing target segments during the equilibrium iterations, the NSUPPRESS feature discussed in Section 4.11.3 can help in obtaining convergence.
- Frictional contact problems using the constraint function algorithm can be sensitive to the choice of sticking velocity (EPST parameter of the CGROUP commands). For most problems, this parameter should be one or two orders of magnitude smaller than the expected sliding velocity. Using an excessively large value leads to a smoother friction law, which generally converges faster but results in smaller frictional forces or more sliding. Using an excessively small value enforces the friction law more accurately but is more likely to experience convergence difficulties.
- Geometric and material nonlinearities can highly depend on the sequence of load application. Thus, for problems involving such features, the load steps should be small. The time step Δt should also be small in dynamic analysis and when time dependent material constitutive relations (e.g., creep) are used.
- The frictionless friction delay feature (Section 4.19.6) can sometimes lead to better convergence since friction will only act once a converged contact solution is established. This feature is

also very useful for many problems involving initial penetrations. In this case, the first time step during which these initial penetrations are removed will be frictionless.

- The stabilized friction delay feature (Section 4.19.6) is useful in problems in which relative sliding is not expected to occur between the contact surfaces, for example, bolted assembly models, since the additional tangential stiffness stabilizes the tangential motion of the contact surfaces.
- The TENSION-CONSISTENT feature (Sections 4.18.4 and 4.18.6) should be used only to obtain the expected uniform traction results in problems with uniform tractions in higher-order elements. Whenever either the normal tractions or the frictional tractions are not expected to be uniform, the TENSION-CONSISTENT feature should not be used.
- When using the multilinear, XEXP and power law contact pressure functions, the contact tolerance RCTOL should be very tight, in order to prevent premature convergence of the solution.

4.27.1 Improperly supported bodies

Many static problems depend on contact to provide the boundary conditions necessary for a stable problem (one in which there are no rigid body modes). Some examples are shown in Fig. 4.27-1.

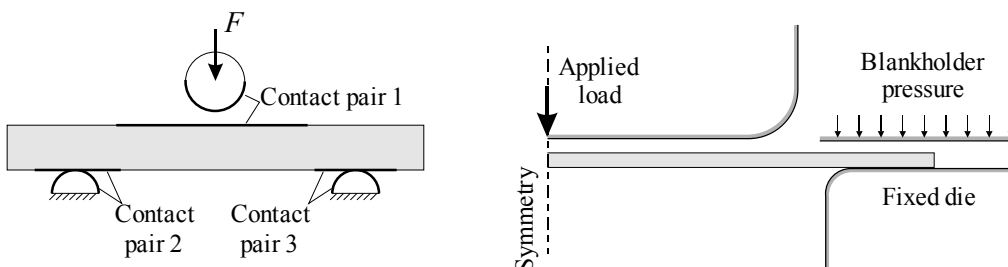


Figure 4.27-1: Examples of improperly supported bodies

In such cases, the stiffness matrix is singular if there is no contact. Even if there is contact the stiffness matrix is still not positive definite. The problem is more serious if natural boundary conditions are applied (forces, moments, or pressure). Weak springs can be added by the user to make the model stable. However, the selection of appropriate locations and stiffnesses for such springs may not be feasible. Some models may be better suited for a dynamic or the low-speed dynamics feature. However, in many cases, this too is not a feasible option. Therefore, several other modeling techniques are available in ADINA to handle such problems. These are stiffness stabilization, absolute contact damping and limiting incremental displacements. These techniques can be used separately or combined in the same model.

Stiffness stabilization

This feature provides a stabilizing effect by scaling all diagonal stiffness terms without affecting the right-hand-side load vector. The outcome of each iteration will be affected, but the final converged solution will not be (within the bounds of the convergence tolerances). Since the stabilization constant is non-dimensional, it should always be a small number. Typical values are between 10^{-12} and 10^{-9} .

Stiffness stabilization is discussed in detail in Section 11.15.

Absolute contact damping

Absolute contact damping adds grounded viscous dampers to all contactor and target nodes. Setting the damping to be only at the initial time step is sufficient for some problems such as those in Fig. 4.27-1. When the first time step converges contact must be established and damping will have been removed. This way, the converged solution will be free of any contact damping. Other problems however, require the damping to be constantly present. In this case, ADINA outputs the damping forces at every time step. These forces should be compared with the reactions in order to ensure that damping is not excessive.

The proper value of the damping constants is problem dependent.

If initial contact damping is used to stabilize a problem involving two contact bodies at least one of which is unsupported, with a gap

g between them, and external force R acting in the gap direction, then a good estimate of the damping constants C_N and C_T is one in which the gap is nearly closed in the first iteration. Starting with the dynamic equations of motion (see Equation 7.3-1) and removing the inertial term (static analysis), and the stiffness term (since one or both bodies are initially unsupported), we obtain

$$\mathbf{C}\dot{\mathbf{U}} = \mathbf{R}$$

where \mathbf{C} is the total damping matrix, which in this case is diagonal, and \mathbf{R} is the applied load vector. We can assume the normal and tangential damping constants to be equal and the velocity to be approximately equal to g divided by the time step size Δt . If force-based absolute damping is used, the total damping contribution is the damping constant times the number of contact nodes N on the unsupported contact surface (the top circular body in the example in Fig. 4.27-1), and if traction-based absolute damping is used, the total damping contribution is the damping constant times the area A_s of the unsupported contact surface.

These considerations lead to the following value of the damping constants:

$$\text{Force-based absolute damping: } C_N = C_T = \frac{R\Delta t}{Ng}$$

$$\text{Traction-based absolute damping: } C_N = C_T = \frac{R\Delta t}{A_s g}$$

Note that this is only an estimate, but is frequently an acceptable one.

Absolute contact damping is discussed in detail in Section 4.4.

Limiting maximum incremental displacement

Limiting the maximum incremental displacement per iteration is useful when a load is applied to a body that is not initially in contact. The model at that stage is unstable and even when stiffness stabilization or absolute contact damping is used, the initial displacement can be excessive leading the program away from the converged solution, and thus making the return to the proper solution difficult. Setting the limiting displacement to about the element length size in this case scales down the potentially huge displacement in the first iteration so that the results remain close to the converged solution.

Note that this feature does not stabilize the stiffness matrix, so in many cases it may be necessary to use it together with stiffness stabilization or absolute contact damping or both.

The limiting maximum incremental displacement feature is discussed in detail in Section 7.2.1.

4.27.2 Contact-impact problems

- Oscillations in velocities and accelerations can sometimes be present in implicit dynamic contact analysis especially for high speed impact problems. These oscillations can be reduced by
 - adding compliance to the contact surfaces,
 - using the Bathe time integration method.
- Adding compliance to the contact surface can significantly reduce the numerical oscillations that result from dynamic time integration. In this case, the compliance factor must be selected such that the contact pressures do not cause excessive penetration.
- The Bathe time integration method provides some numerical damping to the high frequency content of the solution, which includes the contact oscillations.

- Post-impact corrections can also be applied, as described in Section 4.19.8. However post-impact corrections are not available in Cat 2 contact.

4.27.3 Problems with many nodes in contact

The equilibrium iterations can converge only when the state of all contactor nodes is known (whether the node is in contact, and whether the node is sticking or sliding in frictional contact). Thus, when the states of the contactor nodes are changing during the equilibrium iterations, the equilibrium iterations cannot converge.

When only a few contactor nodes change state during a time step, the convergence is relatively easy.

However sometimes many contactor nodes might change state during a time step. Then convergence can be difficult, and cutting back the time step might not help. A schematic example is shown in Fig. 4.27-2.

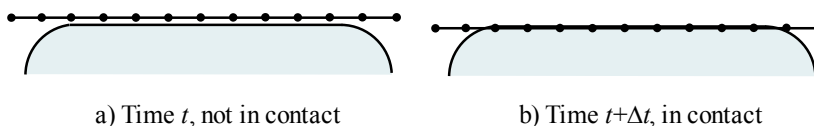


Figure 4.27-2: Many nodes entering contact in a time step

This type of situation is most frequently encountered when solving for an “initial condition”, for example, in metal forming when applying the blank holder force before the motion of the punch. Or in the assembly of parts with bolts.

The number of equilibrium iterations increases as the number of contactor nodes increases. Thus refining the mesh causes the number of equilibrium iterations to increase.

The norms of the out-of-balance forces and other convergence indicators do not decrease, or decrease slowly, for many equilibrium iterations. Then, once the state of the contactor nodes

is known, the indicators decrease suddenly and the solution converges.

Increasing the number of equilibrium iterations allowed can help in obtaining a converged solution.

4.28 Modeling recommendations in explicit analysis

4.28.1 Choice of explicit algorithm

When using Cat 1 contact, either the kinematic constraint or penalty algorithm can be chosen. The penalty algorithm is preferred when both surfaces are rigid or have many fixed or prescribed nodes.

When using Cat 2 contact, only the penalty algorithm can be used.

4.28.2 Recommendations when using the penalty algorithm

- Large oscillations in the contact forces may occur when using the penalty method even though the model is stable. These oscillations can be reduced by adding a penalty rate damping term and/or reducing the penalty stiffness.
- When using the penalty contact algorithm it is important to check that the contact stiffnesses are properly selected. Unduly small penalty stiffnesses will lead to excessive penetrations, while unduly large penalty stiffnesses will lead to excessive oscillations or unstable time integration.
- Large mismatches between the masses of contacting surfaces can lead to problems when using the penalty method. In this case, the normal penalty stiffness required to avoid instability (without reducing the time step) can be unduly small leading to excessive penetrations. The best solution in such cases is to minimize the mismatch by increasing the mass of the rigid surface, or increase the penalty stiffness by setting it manually or by reducing the time step.

4.28.3 Recommendations when using the kinematic constraint algorithm

- Large mismatches between the masses of contacting surfaces should be avoided when using the kinematic constraint method. This mismatch is common when contact involves a rigid surface with a small mass and an applied force, as shown in Fig. 4.28-1. The best solution in such cases is to minimize the mismatch by increasing the mass of the rigid surface.

The inaccuracy in this case results from the way the contact is enforced. The kinematic constraint method first predicts displacements without contact then applies a contact correction. The contact conditions are satisfied more accurately when the penetrations in the predicted configuration are small which is usually the case due to the small time step size of explicit analysis. However, some cases such as that mentioned above lead to large projected penetrations which results in incorrect contact conditions and tensile contact forces.

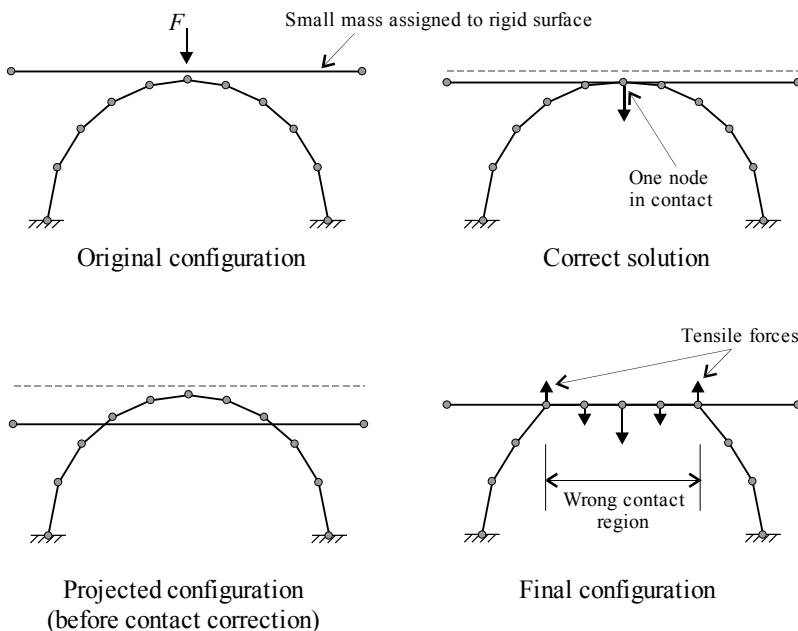


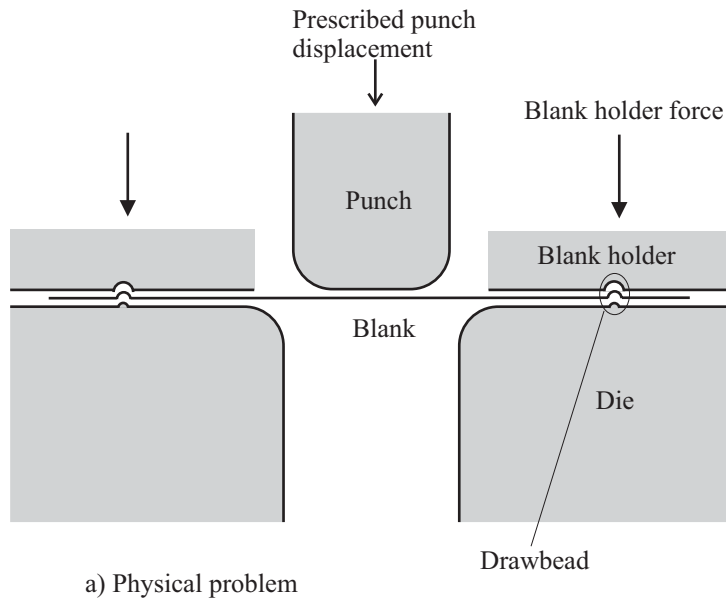
Figure 4.28-1: Performance of kinematic constraint algorithm when contact surfaces have a large mass mismatch.

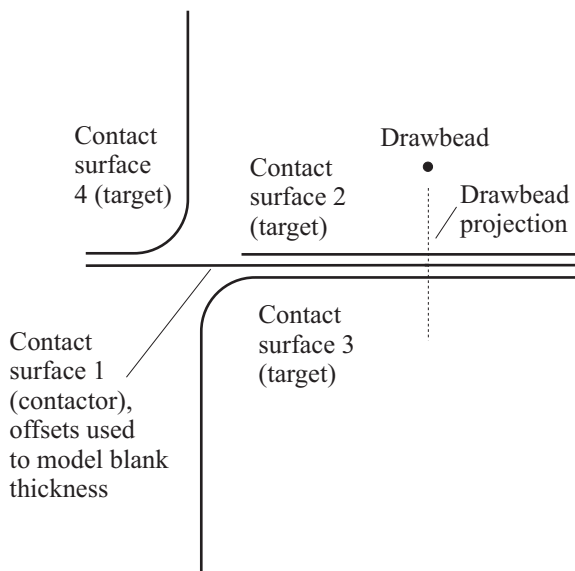
4.29 Rigid-target contact algorithm

4.29.1 Introduction

The rigid target contact algorithm is intended for use in applications in which the target surfaces are considered to be rigid.

Fig. 4.29-1 shows a typical application in metal forming.





b) Modeling with contact surfaces and drawbead

Figure 4.29-1: Sample metal forming analysis using the rigid target contact algorithm

A target surface can either be stationary, can translate as a rigid body or can rotate as a rigid body.

Contact can be frictionless or can include Coulomb friction.

Drawbeads can be included.

2D or 3D contact conditions can be considered.

For 2D contact conditions, the rigid target contact algorithm is the one used in ADINA version 8.3 and earlier.

For 3D contact conditions, the rigid target contact algorithm is completely revised in ADINA version 8.4. However the rigid target contact algorithm of ADINA version 8.3 and earlier is retained in ADINA for backwards compatibility. The revised rigid target contact algorithm of ADINA version 8.4 is the default.

Throughout this section, the rigid target contact algorithm of version 8.3 and earlier is referred to as the “V83” rigid target contact algorithm. This section does not describe the V83 rigid target contact algorithm; for information on the V83 rigid target contact algorithm, see the ADINA Theory and Modeling Guide for version 8.3 (report ARD 05-7).

Models that were set up using the V83 rigid target contact algorithm may need to be revised when using the current rigid target contact algorithm, see the conversion hints in Section 4.29.7.

We suggest that new models not be set up using the V83 rigid target contact algorithm.

It is also possible to solve many problems involving rigid targets using the general contact algorithms described earlier in this chapter. However, the rigid target contact algorithm described here is frequently more effective, because the rigid target contact algorithm uses the assumption of rigid targets to simplify the contact searching.

4.29.2 Basic concepts

4.29.2.1 Contactor surfaces

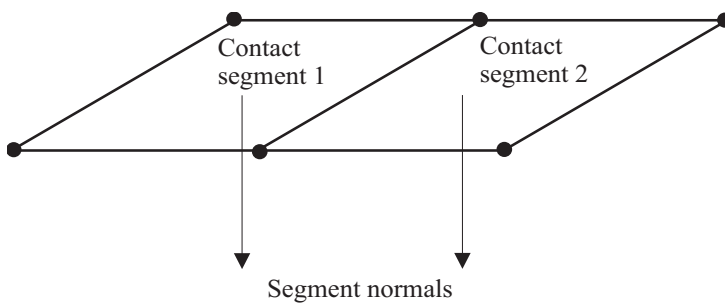
Similar to the other contact algorithms in ADINA, the contact surfaces are organized into contact groups. Each contact surface consists of 3- or 4-node contact segments. A contact pair consists of a contactor surface and a target surface. In the rigid target algorithm, it is allowed for a contactor surface to be in contact with more than one target surface simultaneously.

Contactor surface: The contactor surface definition includes the possibility of offsets.

When there are no offsets specified, the contactor surface is described entirely by the contactor nodes. (Fig. 4.29-2(a)).

When there are offsets specified, the offsets can either be described using spheres centered around the contactor nodes (Fig. 4.29-2(b)), or using the contactor normals (Fig. 4.29-2(c)). In either case, the

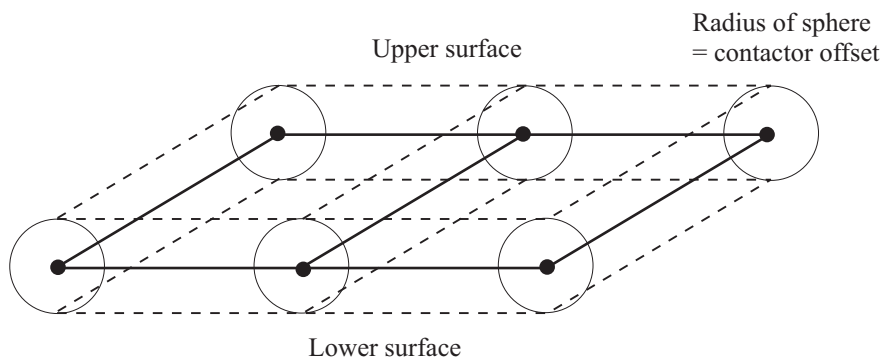
offset magnitude is either constant or taken from the current thickness of attached shell elements, as described below.



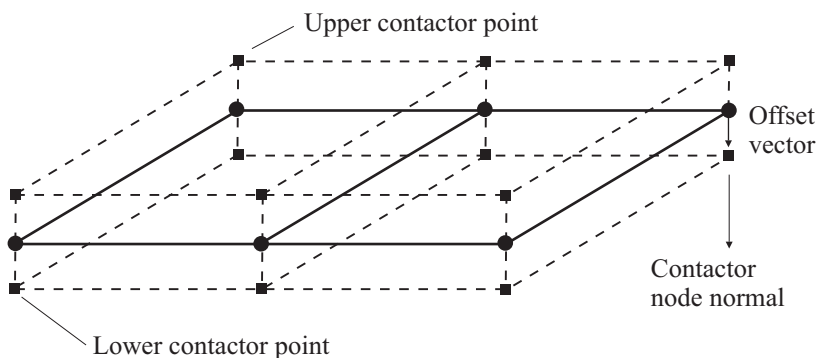
a) Contactor segments without offsets. Segment normals are not used.

Figure 4.29-2: Contacting segments

When the offsets are described using the contactor normals, offset vectors are constructed using the averaged contactor normals and the offset magnitude. The upper and lower contactor points are constructed from the contactor nodes and the offset vectors.



b) Contactor segments with spherical offsets. Segment normals are not used.



Contactor node normal = average of all segment normals

Offset vector = contactor node normal \times offset magnitude

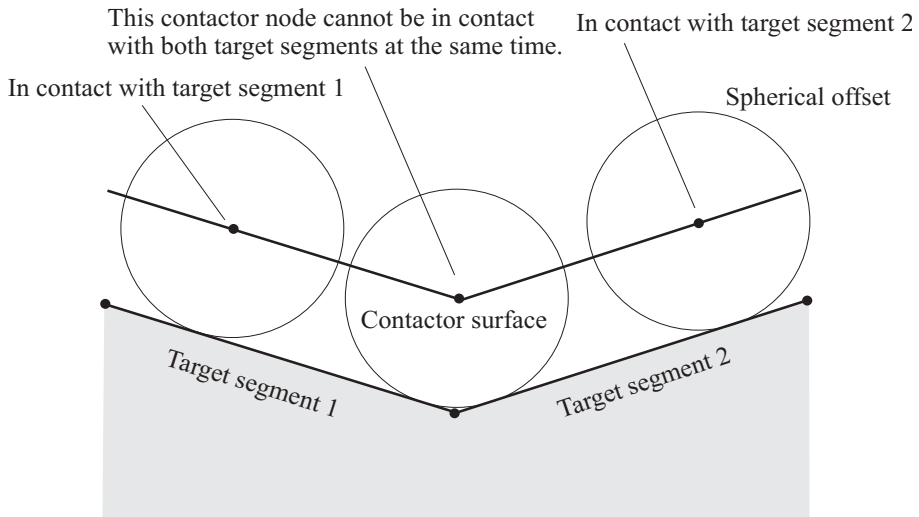
c) Contactor segments with offsets, normals used to describe offsets.

Figure 4.29-2: (continued)

When the target surface is concave, it is possible for the contact situation to be similar to the one shown in Fig. 4.29-3. In this case, when the offsets are described using spheres, the center contactor node cannot be in contact with both target segments at once, hence the center contactor node will oscillate between them. The center contactor node cannot be in contact with target edge 1 since edge 1 is farther away than either of the target segments. Equilibrium iterations in static and implicit dynamic analysis will not converge,

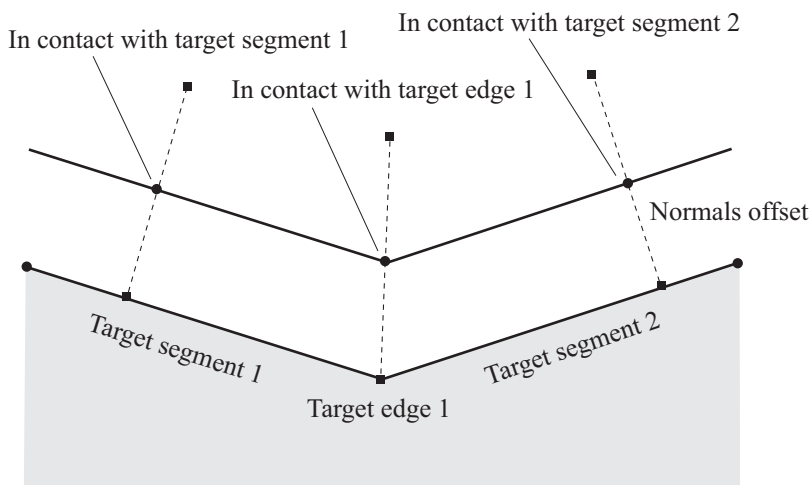
because of the oscillation. However contact is correctly modeled when the offsets are described using normals, because the center contactor node can be in contact with target edge 1.

Contactor and target surfaces viewed from the side for ease of visualization.



a) Spherical offsets, contact incorrectly modeled

Figure 4.29-3: Concave target surface, contactor surface with offsets



b) Normals offsets, contact correctly modeled

Figure 4.29-3: (continued)

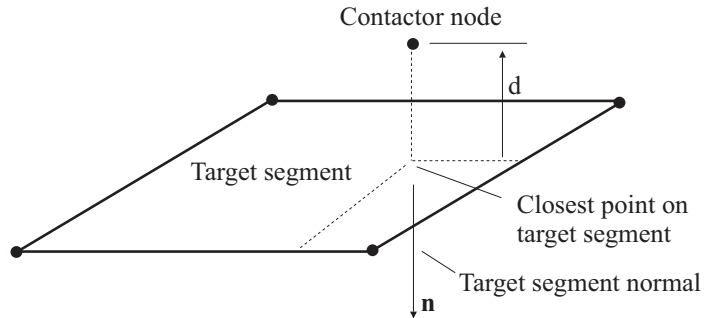
4.29.2.2 Target surfaces

Each target surface is either stationary, or can rigidly move (translate, rotate or a combination of translations and rotations).

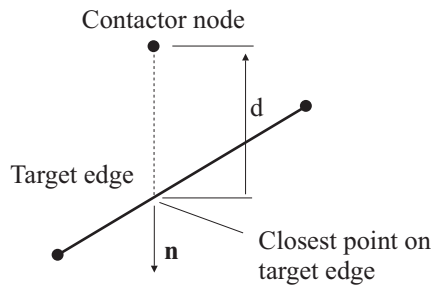
4.29.2.3 Determination of contact between contactor and target

No contactor offsets: It is allowed for a contactor node to be in contact with a target segment, target edge or target node. The program searches for the target segment, edge or node for which the absolute value of the distance d between the contactor node and the target segment, edge or node is minimized, where the distance is measured in the direction opposite to the target normal (Fig. 4.29-4). A positive distance corresponds to a geometric gap; a negative distance corresponds to geometric overlap.

Notice that, for interaction between a contactor node and target edge, or between a contactor node and target node, the target normal direction is taken from the line segment connecting the contactor node and the target, as shown in Fig. 4.29-4.



a) Interaction between contactor node and target segment



b) Interaction between contactor node and target edge

Figure 4.29-4: Interaction between contactor node and target surface

Fig. 4.29-5 shows two target segments with a common target edge. The shaded volumes indicate which of the target entities any contactor node is closest to. Notice that the shaded volume in which the contactor node is closest to the target edge depends upon the angles between the target segments attached to the edge.

Once the target segment, edge or node is determined, then the contact gap is computed using

$$g = d - \text{GAPBIAS}$$

where GAPBIAS can be chosen to, for example, not model contact even if there is geometric overlap. (The default for GAPBIAS is 0.)

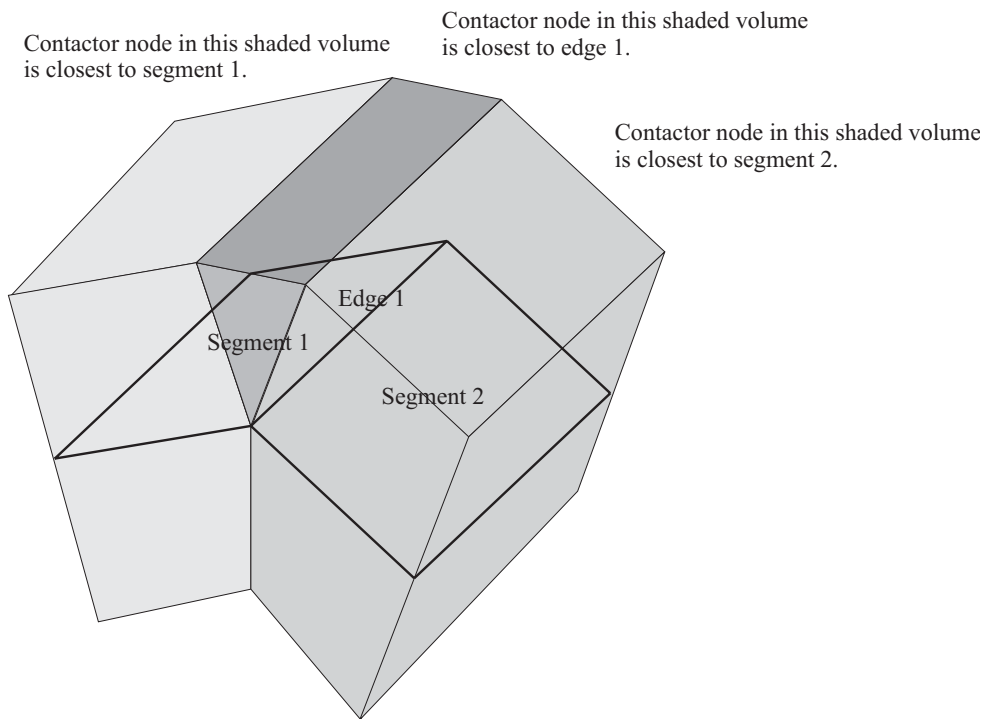


Figure 4.29-5: Interaction of contactor node with target segments and edges

If the corresponding gap is negative, and less than DEPTH, then contact occurs, in other words $-\text{DEPTH} \leq g \leq 0$ is the contact condition. DEPTH can be chosen to limit the depth of the target surface, exactly as in the other contact algorithms.

Contactor offsets described using spheres: In this case, the distance d is determined exactly as if there are no offsets. Then the contact gap is computed using

$$g = d - \text{OFFSET} - \text{GAPBIAS}$$

where $OFFSET$ is the offset magnitude. The process is illustrated for interaction between a contactor node and target segment, assuming that $GAPBIAS = 0$ (Fig. 4.29-6). The same idea is used for interaction between a contactor node and target edge or node.

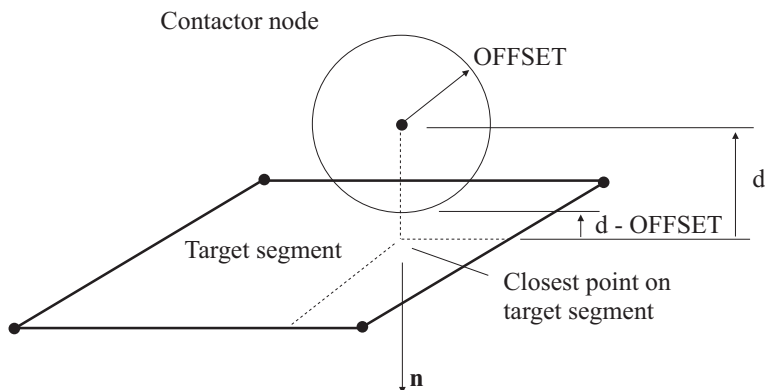


Figure 4.29-6: Interaction between contactor node and target segment, spherical offsets

Contactor offsets described using normals: In this case, contact is detected using the upper and lower contactor points instead of the contactor nodes.

Oscillation checking: The search for the nearest target segment, edge or node is performed every equilibrium iteration in static and implicit dynamic analysis. During the equilibrium iterations, it is possible for the contactor node to move in such a way as to be alternately in contact with two neighboring segments. This is especially true if the target surface is concave. When the contactor node oscillates between two neighboring segments, the solution cannot converge unless oscillation checking is turned on. When oscillation checking is turned on, then, when oscillation is detected between two neighboring segments, the contactor node is placed into contact with the shared target edge. In many cases, this procedure allows the iterations to converge, if in fact the contactor node “should” have been in contact with the shared target edge.

Oscillation checking only forces the contact between the contactor node and shared target edge for the current equilibrium iteration. For the successive equilibrium iterations, the contactor node is

always in contact with the nearest target segment, edge or node. So oscillation checking cannot force contact to the “wrong” target segment, edge or node in a converged solution.

Contact normal force: The normal force corresponding to contact is computed as $F_n = -k_n g$ where the normal force acts in the direction opposite to the target normal direction (Fig. 4.29-7). k_n is the contact normal stiffness, entered as a parameter (see Section 4.29.3 for hints about choosing k_n). k_n can be considered to be a penalty parameter.

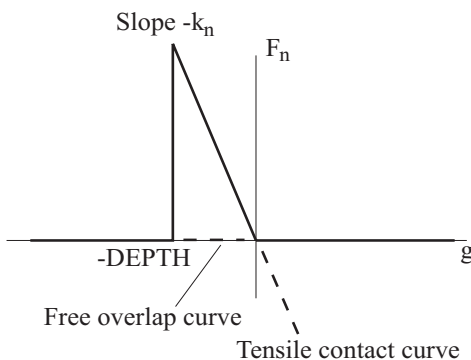


Figure 4.29-7: Normal contact force vs gap

Tensile contact and free overlap: During equilibrium iterations in static and implicit dynamic analysis, the convergence can be affected by the use of two related features: tensile contact and free overlap. These features can be turned on or off in the model input, see Section 4.29.3. The default is to use tensile contact and not to use free overlap.

Tensile contact: The basic ideas for tensile contact are illustrated in Fig. 4.29-8 to 4.29-10.

Fig. 4.29-8 shows the iteration history when tensile contact is not used. For iteration ite-2, the contactor node and target segment overlap. Hence contact is assumed between the contactor node and target segment. For iteration ite-1, because of the relative motion of the contactor node and target segment, the contactor node and

target segment do not overlap. For this iteration, no contact is assumed between the contactor node and target segment. For iteration ite , there is a large overlap because the contactor spring unloads, since there are no forces acting on the contactor spring, and the target does not provide any stiffness to the contactor node. This large overlap causes large contact forces, which can cause trouble in convergence in the successive iterations.

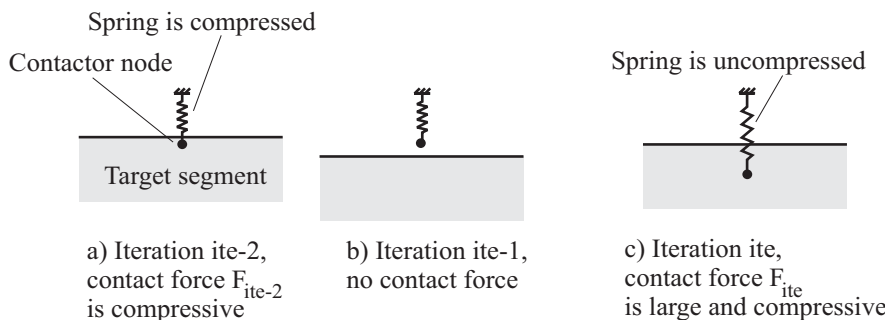


Figure 4.29-8: Iterations when tensile contact is not used

Fig. 4.29-9 shows the iteration history when tensile contact is used.

Now, in iteration $ite-1$, tensile contact is assumed between the contactor node and target segment. In tensile contact; the target surface still provides stiffness to the contactor node. Hence the overlap in iteration ite is small.

Fig. 4.29-10 shows the iteration history when tensile contact is used, and the gap is large. In iteration $ite-1$, tensile contact is assumed, and in iteration ite , no contact is assumed.

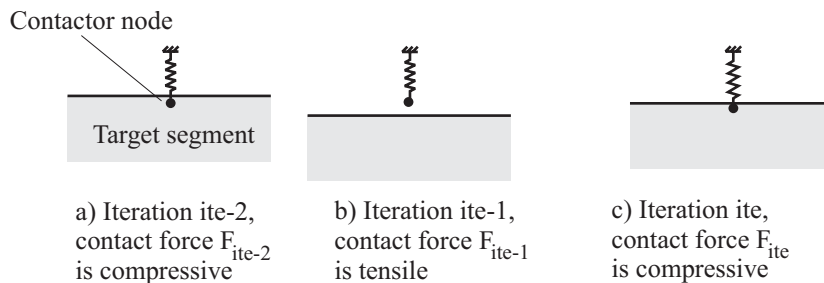


Figure 4.29-9: Iterations when tensile contact is used

It is seen that tensile contact speeds up the convergence when the converged solution is in contact, and slows down the convergence when the converged solution is not in contact.

It is not permitted for an iteration in which tensile contact is present to converge, unless the tensile forces are all less than the value of a user-input parameter (see Section 4.29.3). Hence the use of the tensile contact feature does not affect the converged solution.

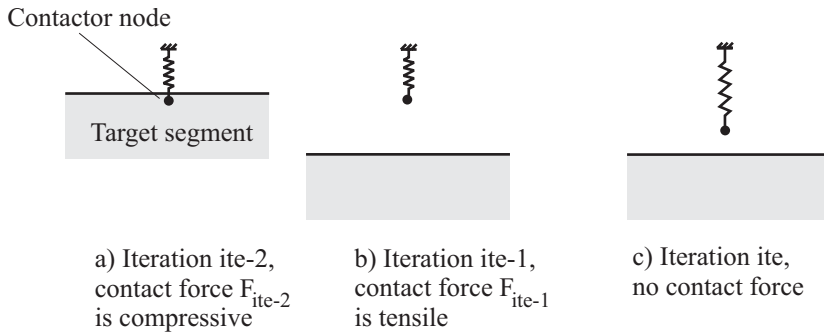


Figure 4.29-10: Iterations when tensile contact is used, converged solution not in contact

Free overlap: The basic ideas for free overlap are illustrated in Fig. 4.29-11 to 4.29-13.

Fig. 4.29-11 shows the iteration history when free overlap is not used. For iteration ite-3, there is a gap between the contactor node and target segment. Hence no contact is assumed between the contactor node and target segment. For iteration ite-2, because of the relative motion of the contactor node and target segment, the contactor node and target segment overlap. For this iteration, contact is assumed between the contactor node and target segment. For iteration ite-1, the contactor spring is stretched because there is a large stiffness between the contactor and target due to the contact. There is a small gap between the contactor and target, hence no contact is assumed. For iteration ite, the contactor spring is unloaded, and there is a gap between the contactor and target.

Fig. 4.29-12 shows the iteration history when free overlap is used. Now, in iteration ite-2, free overlap is assumed between the contactor node and target segment. Hence there is no contact

assumed for this iteration. In iteration ite-1, the contactor spring is unaffected by the motion of the target.

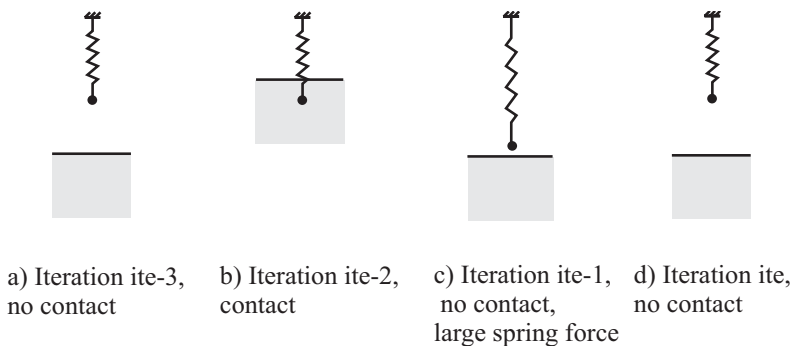


Figure 4.29-11: Iterations when free overlap is not used

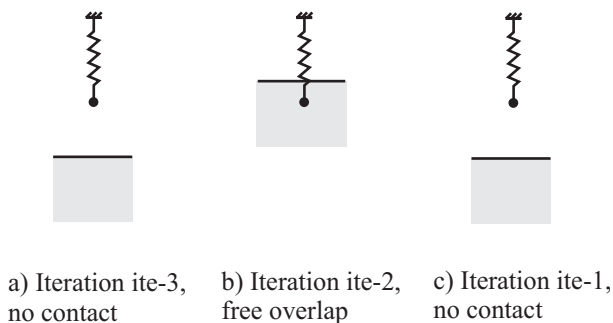


Figure 4.29-12: Iterations when free overlap is used

Fig. 4.29-13 shows the iteration history when free overlap is used, and the overlap remains between target and contactor. In iteration ite-2, free overlap is assumed, and in iteration ite-1, contact is assumed.

It is seen that free overlap speeds up the convergence when the converged solution is not in contact, and slows down the convergence when the converged solution is in contact.

It is not permitted for an iteration in which free overlap is present to converge. Hence the use of the free overlap feature does not affect the converged solution.

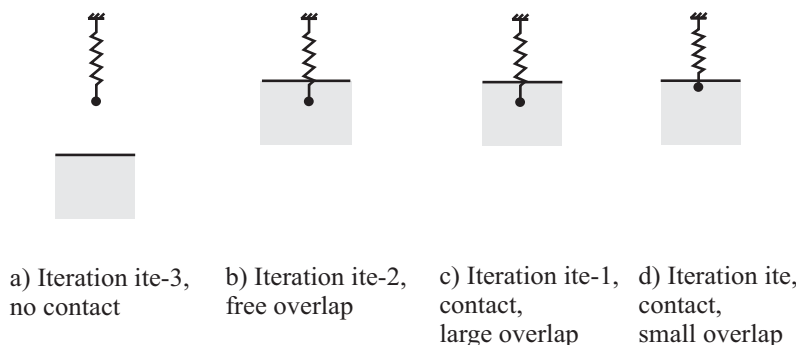


Figure 4.29-13: Iterations when free overlap is used

4.29.2.4 Frictional contact

The friction force is calculated using the relative sliding velocity between the target and contactor. The relative sliding velocity $\dot{\mathbf{u}}_f$ is calculated from the velocities of the contactor and target using

$$\dot{\mathbf{u}}_f = (\dot{\mathbf{u}}_c - \dot{\mathbf{u}}_t) - ((\dot{\mathbf{u}}_c - \dot{\mathbf{u}}_t) \cdot \mathbf{n})\mathbf{n}$$

where $\dot{\mathbf{u}}_c$ is the velocity of the contactor node, $\dot{\mathbf{u}}_t$ is the velocity of the target and \mathbf{n} is the target normal.

In static analysis, the contactor and target velocities are calculated using the nodal incremental displacements divided by the time step. In dynamic analysis, the contactor and target velocities are taken from the nodal velocities.

The friction force magnitude is computed using

$$\begin{aligned} |F_f| &= \frac{\|\dot{\mathbf{u}}_f\|}{\dot{u}_{f\min}} \mu F_n, \quad \|\dot{\mathbf{u}}_f\| < \dot{u}_{f\min} \\ &= \mu F, \quad \|\dot{\mathbf{u}}_f\| \geq \dot{u}_{f\min} \end{aligned}$$

where F_n is the normal contact force and μ is the Coulomb

friction constant (Fig. 4.29-14). $\dot{u}_{f\min}$ is the minimum sliding velocity, entered as a parameter (see Section 4.29.3 for hints about choosing $\dot{u}_{f\min}$). The direction of the friction force is always opposite to \dot{u}_f .

When $\|\dot{u}_f\| < \dot{u}_{f\min}$, the friction is sticking, otherwise the friction is sliding.

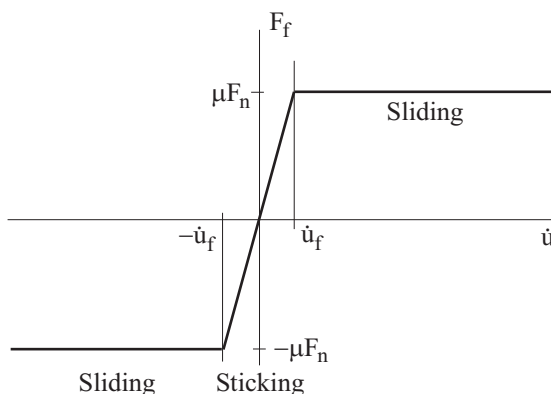


Figure 4.29-14: Friction force vs velocity

Oscillation checking with friction: During equilibrium iterations in static or implicit dynamic analysis, it is possible for the contactor node to undergo “sliding reversals”. Namely, the contactor node slides in one direction for an equilibrium iteration, then reverses sliding direction for the next equilibrium iteration. When sliding reversals occur, the solution cannot converge unless oscillation checking is turned on. When oscillation checking is turned on, then, when sliding reversals are detected, the contactor node is placed into sticking contact, even if the sticking force is larger than the sliding force, and convergence is prevented for the current equilibrium iteration.

Oscillation checking only forces sticking friction for the current equilibrium iteration. For successive equilibrium iterations, the frictional state (sliding or sticking) is determined as usual from the sliding and sticking forces. So oscillation checking cannot

converge to a solution in which the frictional state is wrong.

4.29.2.5 Drawbeads

The rigid target contact algorithm includes a simplified drawbead model. Fig. 4.29-15 shows a diagram of a drawbead, as seen from the side, along with the corresponding drawbead model definition and drawbead tractions.

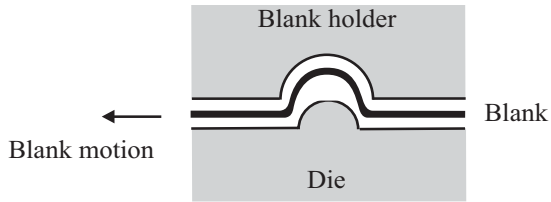
Fig. 4.29-16 shows the drawbead model in three dimensions. In this drawbead model, a drawbead is modeled as a series of connected drawbead segments, in which each drawbead segment connects the successive nodes used to model the drawbead. The drawbead segments are projected in the global z direction, and the projected drawbead segments onto the contactor surface, target surface 1 and target surface 2 are used in the drawbead calculation. Note that as the contactor surface moves, the projected drawbead segments will interact with different contactor segments.

Restrictions: The two target surfaces should not rotate or translate relative to each other in such a way as to cause relative sliding between the two target surfaces. Also the two target surfaces should not rotate or translate so as to cause the projection of the drawbead segments onto the target segments to change. Hence, the two target surfaces can only translate in the z direction.

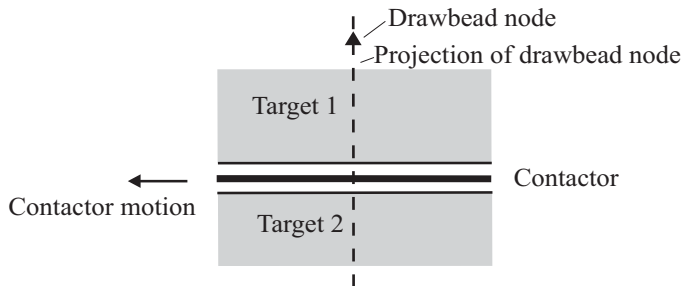
Consistent with these assumptions, the drawbead velocity is taken from the absolute velocity of the contactor surface underneath the projected drawbead segments.

The elements attached to the contactor surface are assumed to be 3- or 4-node shell elements.

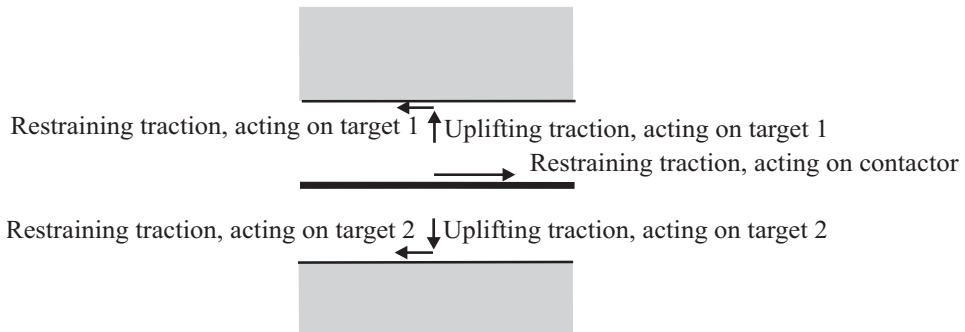
Activation of drawbead: If the distance between the target surfaces, minus the thickness of the contactor, is less than the drawbead height, then the drawbead is active (Fig. 4.29-17). The thickness of the contactor is taken as twice the offset of the contactor. The drawbead height is a user-input parameter. If the drawbead is not active, it does not apply restraining and uplifting tractions to the model.



a) Drawbead diagram



b) Drawbead model definition



c) Drawbead tractions

Figure 4.29-15: Drawbead concepts

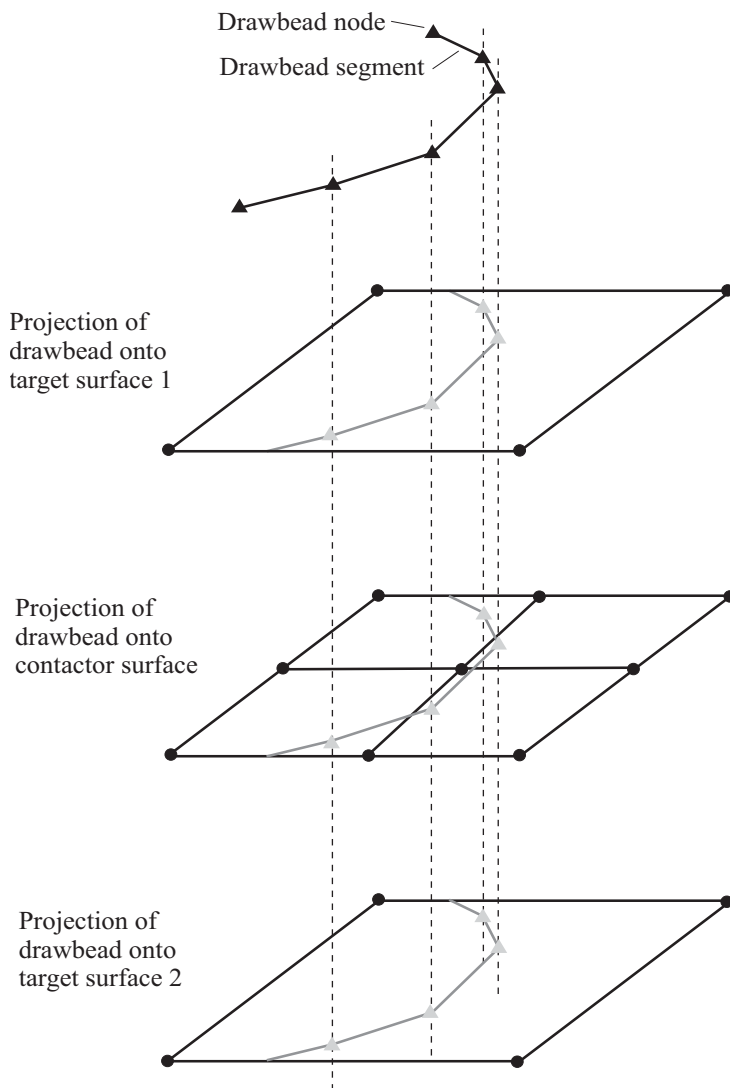


Figure 4.29-16: Specification of a drawbead

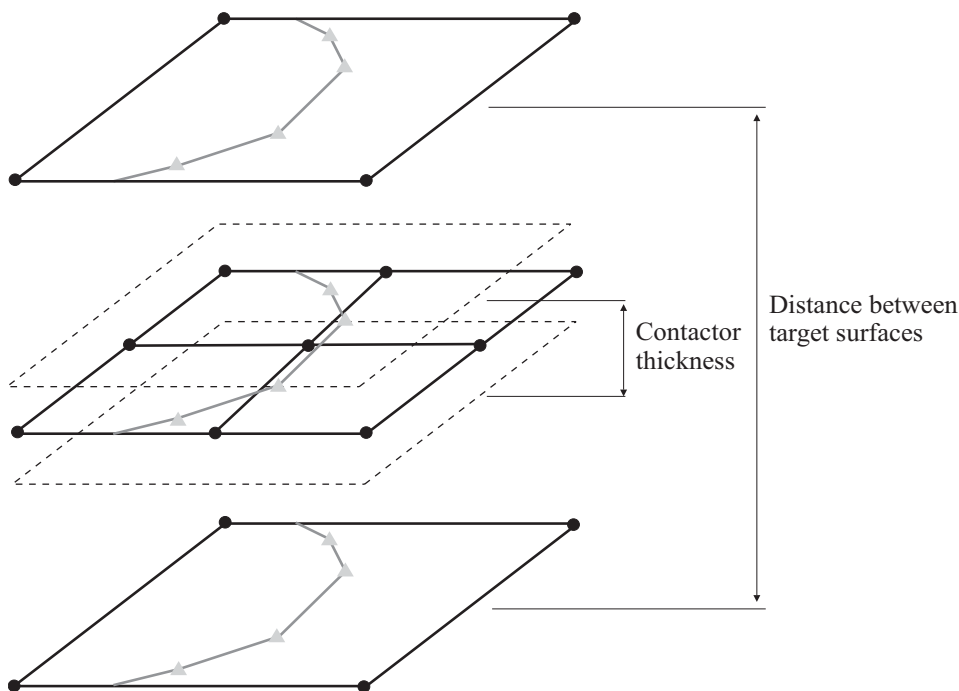


Figure 4.29-17: Activation of a drawbead

Drawbead restraining tractions: The magnitude of the restraining traction acting onto the contactor surface is computed from the velocity of the contactor surface underneath the projected drawbead segments. The formula is

$$|T_R| = \frac{|\dot{u}_{DB}|}{\dot{u}_{DB\min}} T_{RMAX}, \quad |\dot{u}_{DB}| < \dot{u}_{DB\min}$$

$$= T_{RMAX}, \quad |\dot{u}_{DB}| \geq \dot{u}_{DB\min}$$

as shown in Fig. 4.29-18. $\dot{u}_{DB\min}$ is the minimum drawbead sliding velocity and T_{RMAX} is the restraining traction corresponding to sliding. Both of these values are user-input. \dot{u}_{DB} is the velocity of the contactor surface underneath the projected drawbead segments, measured normal to the drawbead segments, but in the plane of the contactor. T_R is the restraining traction (force per unit length), and

always acts in the direction opposite to \dot{u}_{DB} (Fig. 4.29-19).

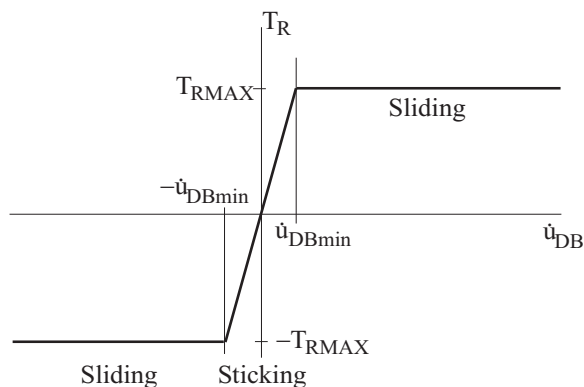


Figure 4.29-18: Drawbead restraining traction vs velocity

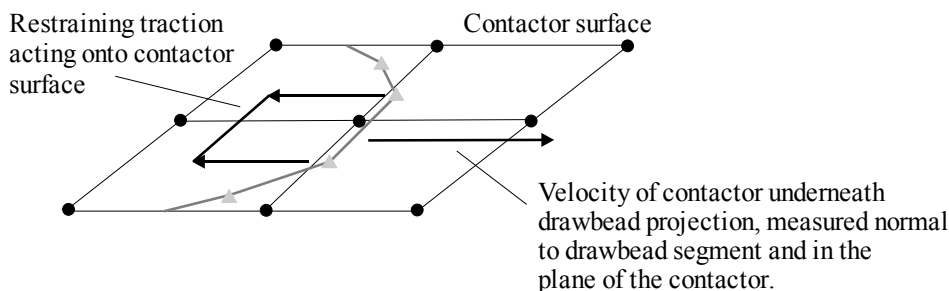


Figure 4.29-19: Drawbead restraining traction

The restraining traction is applied to the contactor, and tractions equal and opposite to half of the restraining traction are applied to each of the targets (Fig. 4.29-20).

Drawbead restraining tractions are like friction forces in that the restraining tractions always oppose the contactor velocity, and that for contactor velocities higher than \dot{u}_{DBmin} , the restraining tractions no longer increase. For contactor velocities lower than \dot{u}_{DBmin} , the drawbead acts as a damper, with damping constant equal to

$\frac{T_{RMAX}}{\dot{u}_{DBmin}}$. (See Section 4.29.3 for hints about choosing \dot{u}_{DBmin} .)

Drawbead uplifting tractions: The drawbead also applies uplifting tractions T_U (force per unit length) to each of the target surfaces:

$$T_U = \frac{|T_R|}{T_{RMAX}} T_{UMAX}$$

T_{UMAX} is a user-input drawbead property. $T_{UMAX} = 0$ is allowed. The uplifting traction always creates forces that tend to separate the target surfaces (Fig. 4.29-20).

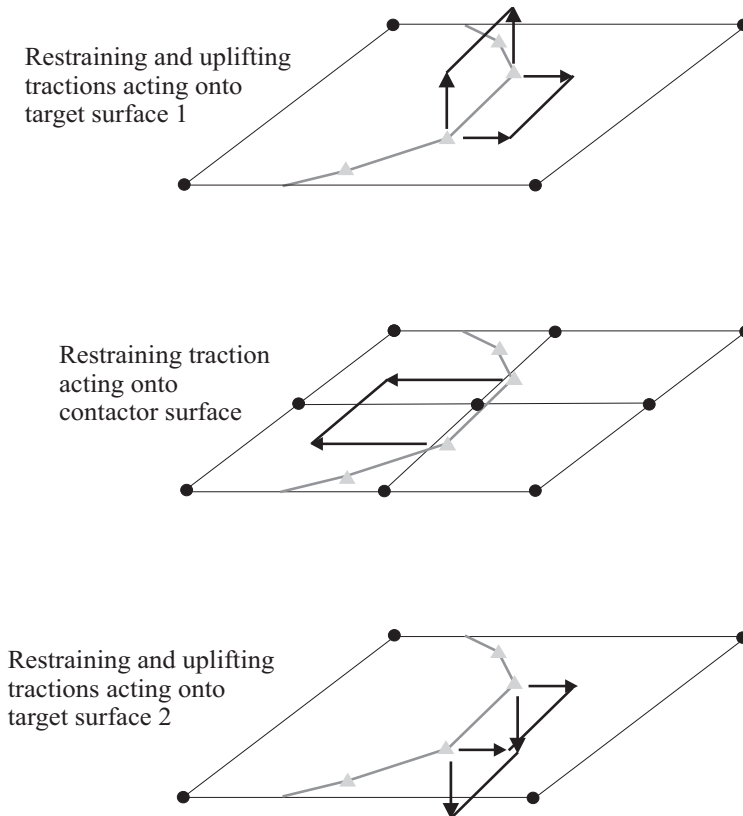


Figure 4.29-20: Restraining and uplifting tractions

4.29.3 Modeling considerations

Algorithm used: The current rigid target contact algorithm is selected by default.

To select the rigid target contact algorithm of version 8.3, use the command

```
CONTACT-CONTROL ... RT-ALGORITHM=V83
```

Modeling of target surfaces: If the target surface translates or rotates, all of the nodes on the target surface must be connected to a “master node”, either using constraint equations or using rigid links. For example, in Fig. 4.29-21, all of the nodes on the lower target surface are connected to a master node using rigid links.

It is not allowed for the nodes on a target surface to have independent degrees of freedom. All degrees of freedom for the nodes on a target surface must be fixed or constrained.

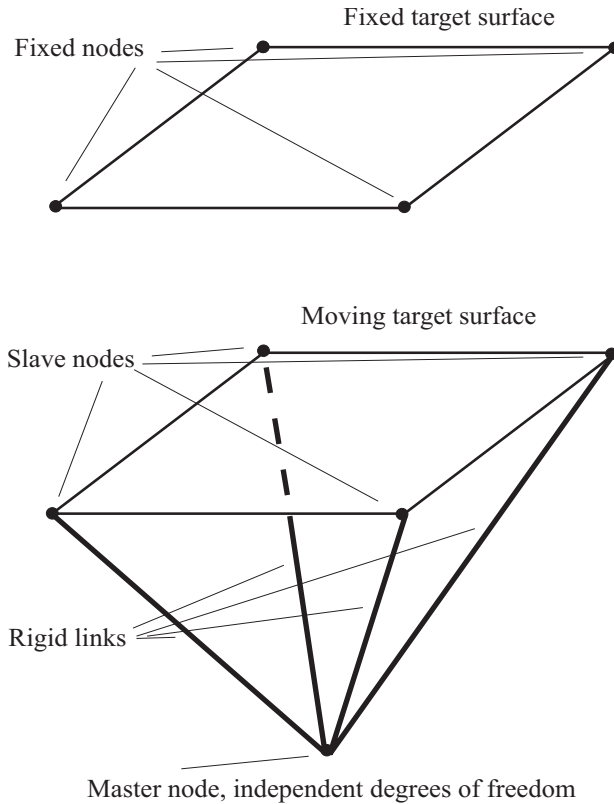


Figure 4.29-21: Modeling of fixed and moving target surfaces

Modeling of contactor surfaces: The amount and description of offset is determined by the CGROUP parameters

OFFSET-TYPE={NO, CONSTANT, TRUE}

OFFSET-DETECT={AUTOMATIC, SPHERES, NORMALS}

and

OFFSET= (constant offset, used only if OFFSET-TYPE=CONSTANT)

OFFSET-DETECT=AUTOMATIC means that ADINA determines

the offset description (SPHERES or NORMALS). The criterion used by ADINA is that an offset description of SPHERES is used for each target surface that is convex or flat, and an offset description of NORMALS is used for each target surface that is concave.

When normals are used for the offset description, small steps should be used in static and implicit dynamic analysis. This is because the offset vectors are assumed to remain constant during the equilibrium iterations. In particular, at convergence, the offset vectors corresponding to the previous converged solution are used.

Determination of contact, target surface buckets: The target segment, edge or node that minimizes the distance between the contactor node and the target surface is determined. To speed up the search, the target surface volume is partitioned into subvolumes (buckets). ADINA automatically determines the number of buckets, but you might want to override the program's choice. Parameters

CONTACTPAIR ... NX, NY, NZ

allow you to specify the number of buckets in each of the global coordinate directions (=0, ADINA automatically determines the number of buckets). Increasing the number of buckets speeds up the search, but increases memory consumption.

Determination of contact, modeling issues: It is possible for the closest target segment, edge or node to not be the expected one. An example is shown in Fig. 4.29-22. In this example, the rim of the wheel is modeled with target segments. Because the distance between a contactor node and a target segment is measured in the direction of the target segment normal, a contactor node interacts with the lower target surface, and the contact algorithm detects a large overlap between this contactor node and the lower target surface.

Another example is shown in Fig. 4.29-23. In Fig. 4.29-23(a), there is a gap between the contactor node and the closest target segment, as expected. In Fig. 4.29-23(b), the punch has moved upward relative to the contactor node. Now there is a large overlap

between the contactor node and the closest target segment. This segment is the only segment with a normal that points in the direction of the contactor node.

In both Fig. 4.29-22 and Fig. 4.29-23, the large overlap is unintended. In static and implicit dynamics, the equilibrium iterations would most likely not converge. In explicit analysis, the large forces between contactor and target would overdistort the elements attached to the contactor node.

One way to avoid the large overlaps is to use the DEPTH feature so that contact is not detected between the contactor node and the incorrect target segment. Another way to avoid this issue is to create additional target segments as shown. Then the contactor node is closest to one of the additional target segments.

Choice of k_n : The default value of the normal contact stiffness k_n is 1E11. However, k_n can be chosen for optimal convergence. Note that increasing k_n causes the maximum overlap between the contactor and the target to become smaller. Also, increasing k_n can lead to convergence difficulties.

We recommend that you use the smallest value of k_n such that the maximum overlap is still acceptably small. For example, if the target surface is curved, there will be a geometric error associated with using a coarse contactor surface (Fig. 4.29-24). There is no advantage if the maximum overlap is less than the geometric error. So, if the mesh is coarse, a large maximum overlap can be used.

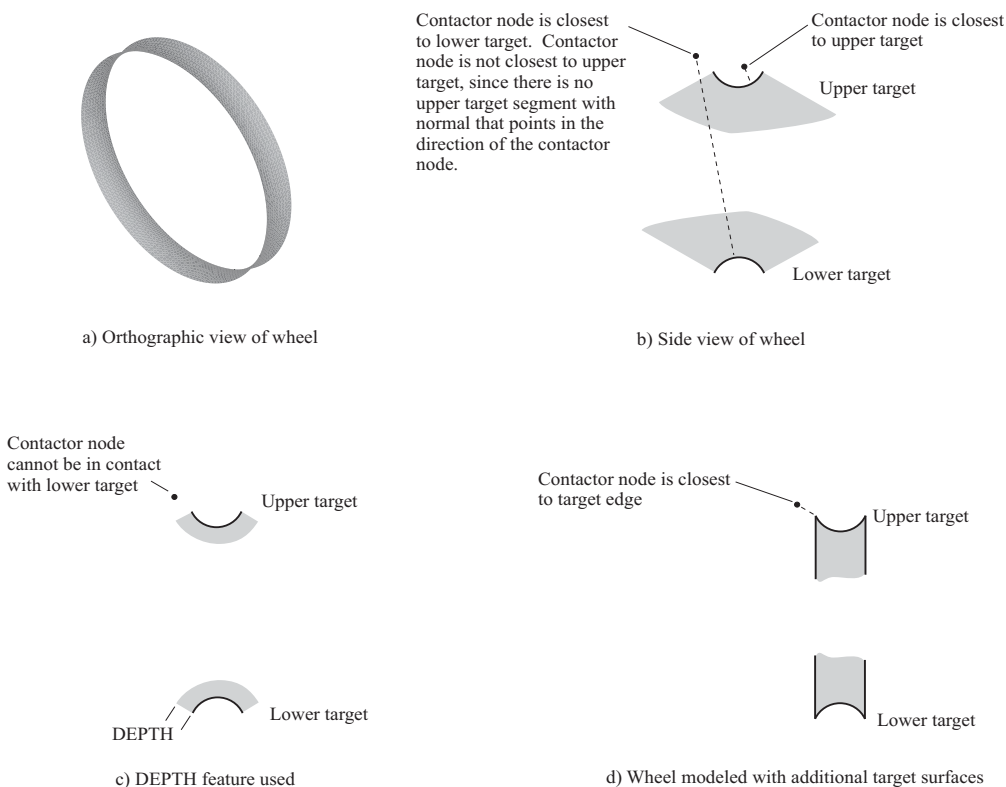


Figure 4.29-22: Meshing of a wheel

Another consideration for the choice of k_n is the following.

Because the rigid target algorithm is node-based, and because the contact stiffness is the same for each node in contact, the stresses computed in higher-order elements on the contactor surface will be inaccurate, if k_n is too small. For example, in a problem involving pressing an element onto a contact surface, k_n should be greater

than $100 \frac{EA}{nL}$ where E is the Young's modulus, A is the contact area, L is the element thickness (in the contact direction) and n is the number of nodes on the contact area. The basic concept is illustrated in Fig. 4.29-25.

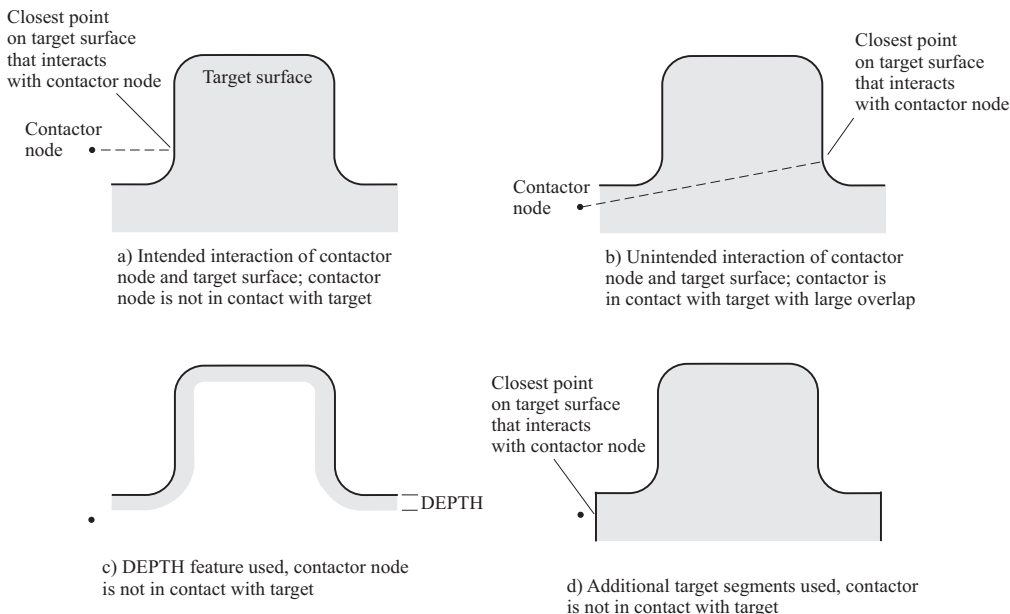


Figure 4.29-23: Modeling of a punch

This issue also arises when lower-order elements are used, but when lower-order elements are used, the variation in the consistent nodal point forces is much less, so k_n can be smaller for the same accuracy in the element stresses.

Explicit dynamics time step size: In explicit dynamics, the time step should be smaller than

$$\Delta t = \sqrt{2} \sqrt{\frac{m}{k_n}}$$

This formula is derived from the following considerations. Consider a single contactor node with mass m and no additional stiffness or damping, with a velocity normal to the target. If this node just touches the target at time $t - \Delta t$, and penetrates the target at time t , the node should remain in contact at time $t + \Delta t$. The choice of Δt given above satisfies this condition.

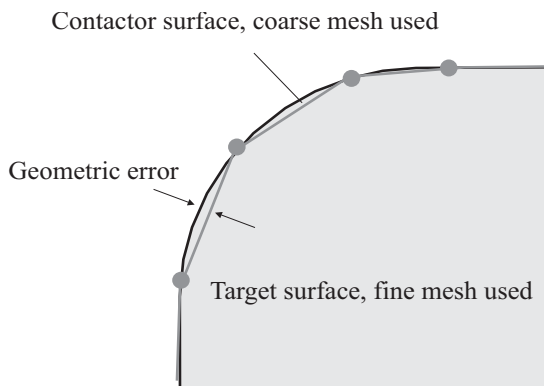


Figure 4.29-24: Modeling of a curved target surface

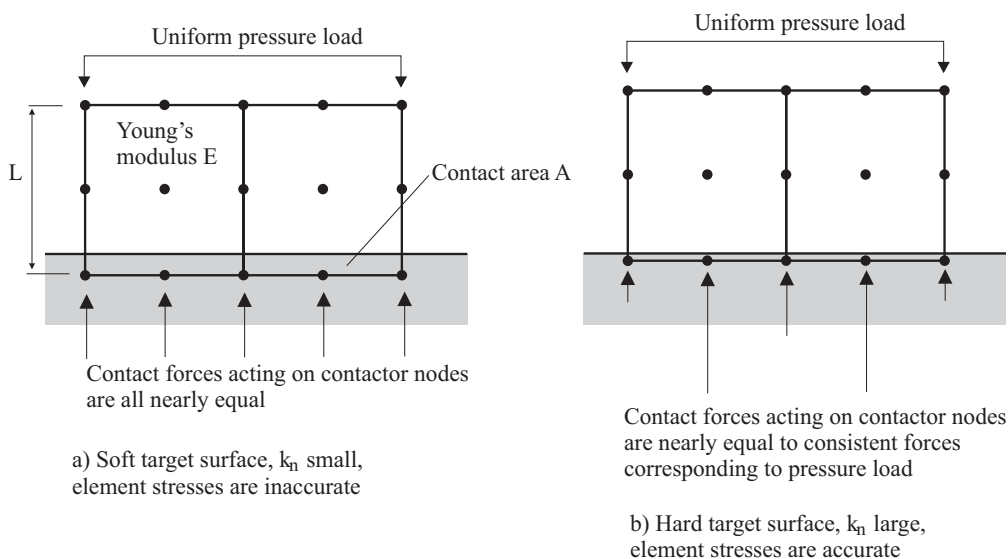


Figure 4.29-25: Higher-order elements and rigid target contact

Clearly, decreasing k_n will increase the time step Δt .

A node that is out of contact at time $t - \Delta t$, in contact at time t and out of contact at time $t + \Delta t$ is said to have had a contact reversal (Fig. 4.29-26).

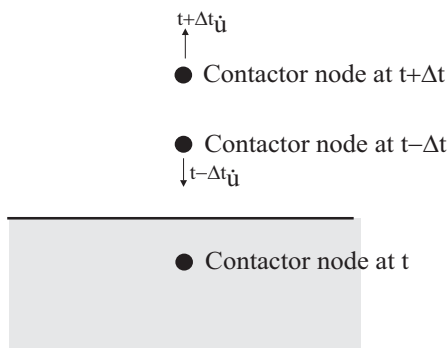


Figure 4.29-26: Contact reversal due to too large time step in explicit dynamics

Time step selection in frictional contact: The time step size will affect the frictional velocities and hence the results. This is because, in static analysis, the nodal velocities used in the friction calculations are calculated as the incremental displacements divided by the time step.

In those steps of the analysis in which friction is important, a “realistic” time step should be used.

In those steps of the analysis in which friction is not important, a large time step can be used, which causes the velocities to be small. For example, in metal forming analysis, a large time step size can be used when establishing the blank holder force, and during springback calculations.

Choice of $\dot{u}_{f\min}$ for frictional contact: The default value of the minimum sliding velocity $\dot{u}_{f\min}$ is 1E-10. However $\dot{u}_{f\min}$ can be chosen for optimal convergence. Decreasing $\dot{u}_{f\min}$ can lead to convergence difficulties.

We recommend that you choose $\dot{u}_{f\min}$ either from experimental data, or use the largest value of $\dot{u}_{f\min}$ acceptable in your application.

Explicit dynamics time step size for frictional contact: In explicit dynamics, the time step should be smaller than

$$\Delta t = \frac{2m\dot{u}_{f\min}}{\mu\|F_n\|}$$

to prevent reverse sliding. This formula is derived from the finite difference equation corresponding to explicit time integration, when applied to a single contactor node with mass m and no additional stiffness or damping, sticking to the target, but with a nonzero sticking velocity. If Δt is larger than the value given above, the velocity will increase, and eventually the node will slide. The sliding will then tend to “reverse”, that is, for a given time step, the sliding direction will be opposite to the sliding direction in the previous step (Fig. 4.29-27)

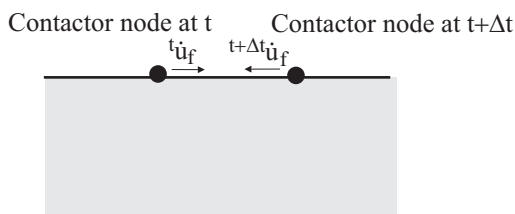


Figure 4.29-27: Reverse sliding due to too large time step in explicit dynamics

Note that when the time step is greater than $\Delta t = \frac{2m\dot{u}_{f\min}}{\mu\|F_n\|}$, the solution is still stable.

Choice of $\dot{u}_{DB\min}$ for drawbeads: The default value of the drawbead minimum sliding velocity $\dot{u}_{DB\min}$ is 1E-8. However $\dot{u}_{DB\min}$ can be chosen for optimal convergence. Decreasing $\dot{u}_{DB\min}$ can lead to convergence difficulties. We recommend that you choose $\dot{u}_{DB\min}$ either from experimental data, or use the largest value of $\dot{u}_{DB\min}$ acceptable in your application.

Explicit dynamics time step size for drawbeads: In explicit dynamics, the time step should be smaller than

$$\Delta t = \frac{2m_{DB}\dot{u}_{DB\min}}{T_{RMAX}}$$

(similar to the time step for friction). Here m_{DB} is the contactor mass/unit length.

Automatic time step selection in explicit dynamics: When using the automatic time step selection options in explicit dynamics, the time step returned from the rigid target contact algorithm is

$$\Delta t = \min_i \sqrt{\frac{m_i}{k_n}}$$

where the minimum is taken over all contactor nodes. Notice that friction is not considered in the automatic time step selection; this is because the model remains stable even if the time step is larger than the friction time step discussed above.

Other user-input parameters:

CGROUP ... TENS-CONTACT={YES/NO}

This controls the use of the tensile contact feature discussed above. The default is YES. Tensile contact is not used in explicit dynamics.

The tensile contact feature is especially useful when shell offsets are used and/or friction is present.

CGROUP ... FREE-OVERLAP={YES/NO}

This controls the use of the free overlap feature discussed above. The default is NO. Free overlap is not used in explicit dynamics.

CGROUP ... GAP-PUSH

When GAP-PUSH is zero, the feature is not used (the default).
When GAP-PUSH is greater than zero, then, when the overlap is less than GAP-PUSH, the contact stiffness is adjusted so that the contactor node is pushed to a gap of GAP-PUSH (in the absence of additional stiffness on the contactor node.) The GAP-PUSH feature is not used in explicit dynamics.

The intent of the GAP-PUSH feature is to accelerate the separation of the contactor and target. Occasionally this can lead to faster convergence of the equilibrium iterations.

CGROUP ... TENSILE-FORCE

The maximum tensile force for a node in tensile contact for which convergence is allowed (default value 0.001). All nodes in tensile contact must have a tensile force less than this value for the solution to converge.

CGROUP ... OSCILLATION-CHECKING

If OSCILLATION-CHECKING = 0, then oscillation checking (described above) is turned off. If OSCILLATION-CHECKING = ITE > 0, then oscillation checking is activated after equilibrium iteration ITE. The default is 5. OSCILLATION-CHECKING is not used in explicit dynamics.

4.29.4 Rigid target contact reports for static and implicit dynamics

The following messages are output at the end of each converged solution.

Maximum overlap at convergence:

Meaning: Self-explanatory

Recommend: If the maximum overlap is too large, increase k_n ;
if the maximum overlap is too small, decrease k_n .

Maximum tensile contact gap during iterations for nodes in contact at convergence:

Meaning: A node that is in contact at the start of the time step may temporarily move out of contact during the iterations, then go back into contact before convergence. This report item reports the maximum contact gap of all such nodes. When the tensile contact gap is large, then convergence may be difficult.

Recommend: Either reduce the time step or decrease k_n to reduce the tensile contact gap.

Maximum friction velocity at convergence:

Meaning: For nodes in frictional contact, this is the maximum friction velocity of a node (either sticking or sliding).

Recommend: If the maximum velocity is less than $\dot{u}_{f\min}$, and the corresponding node should be sliding, decrease $\dot{u}_{f\min}$.

Number of nodes in contact, number of nodes in sticking contact, number of nodes in sliding contact:

Meaning: Self-explanatory. Each node is counted once for each target surface that the node is in contact with. So a node that is in contact with two target surfaces simultaneously is counted twice.

Change of contact status during iterations:

Meaning: The number of nodes that switch contact status (not in contact to in contact, or vice versa), is reported. If there are many nodes that switch contact status, this may cause convergence difficulties.

Recommend: Either reduce the time step or decrease k_n

In contact at convergence, in tensile contact during iterations.

Meaning: The number of nodes which were in tensile contact during the iterations (meaning that the nodes were almost out of

contact) and in contact in the converged solution. When there are many such nodes then convergence may be difficult.

Recommend: Either reduce the time step or decrease k_n to reduce the likelihood that nodes go into tensile contact.

Change in frictional contact status during iterations:

Meaning: The number of nodes that change frictional contact status (from sticking to sliding or vice versa) is reported.

Recommend: If there are many nodes that switch frictional contact status, reduce the time step or increase $\dot{u}_{f\min}$.

The following messages are output at the end of each solution that did not converge.

Maximum change of contact force at end of iterations:

Meaning: The contactor node for which the contact force had the largest change is output.

Recommend: Examine the model near that contactor node for hints about why the solution did not converge.

Change of contact status at end of iterations:

Meaning: The number of nodes that are changing contact status at the end of the iterations.

Recommend: Reduce the time step or decrease k_n .

Sliding reversal at end of iterations:

Meaning: The number of nodes that are undergoing sliding reversals at the end of the iterations.

Recommend: Reduce the time step or increase $\dot{u}_{f\min}$

Change of target entity at end of iterations.

Meaning: The number of nodes that are oscillating between different target entities at the end of the iterations.

Recommend: If you do not have oscillation checking turned on, turn it on. Otherwise refine the target surfaces, or reduce the time step.

4.29.5 Rigid target contact report for explicit dynamics

The following items are output for each time step in which results are printed or saved:

Number of nodes in contact, number of nodes in sticking contact, number of nodes in sliding contact:

Meaning: See the corresponding message in Section 4.29.4.

Maximum overlap since solution start; maximum overlap since last report:

Meaning: See the corresponding message in Section 4.29.4.

Recommend: See the corresponding recommendations in Section 4.29.4

Maximum friction velocity since solution start, maximum friction velocity since last report:

Meaning: See the corresponding message in Section 4.29.4.

Recommend: See the corresponding recommendations in Section 4.29.4.

Contact reversals since solution start, since last report:

Meaning: This is a count of the total number of contact reversals. Also the number of contact reversals for the node with the most contact reversals is given, along with the mass of the node.

Recommend: To reduce the number of contact reversals, either reduce the time step or decrease k_n

Sliding reversals since solution start, since last report:

Meaning: This is a count of the total number of sliding reversals. Also the number of sliding reversals for the node with the most sliding reversals is given, along with the mass of the node.

Recommend: To reduce the number of sliding reversals, either reduce the time step or increase $\dot{u}_{f\min}$.

Drawbead reversals since solution start, since last report:

Meaning: This is a count of the total number of drawbead reversals. Also the number of drawbead reversals for the drawbead segment with the most drawbead reversals is given.

Recommend: To reduce the number of drawbead reversals, either reduce the time step or increase $\dot{u}_{DB\min}$.

4.29.6 Modeling hints and recommendations

- In statics and implicit dynamics, for a time step in which contact is established over a large area, many equilibrium iterations may be required. This is because the solution cannot converge until the nodes in and out of contact are determined, and it may take many equilibrium iterations to determine which nodes are in and out of contact. An example is shown in Fig. 4.29-28. The ATS cutback method will not be effective for this time step. Rather, you should set the maximum number of iterations very large, so that the program can find the converged solution.
- In statics and implicit dynamics, when forming a part that is relatively thin, setting the “plastic-algorithm” flag to 2 can allow the use of larger time steps.

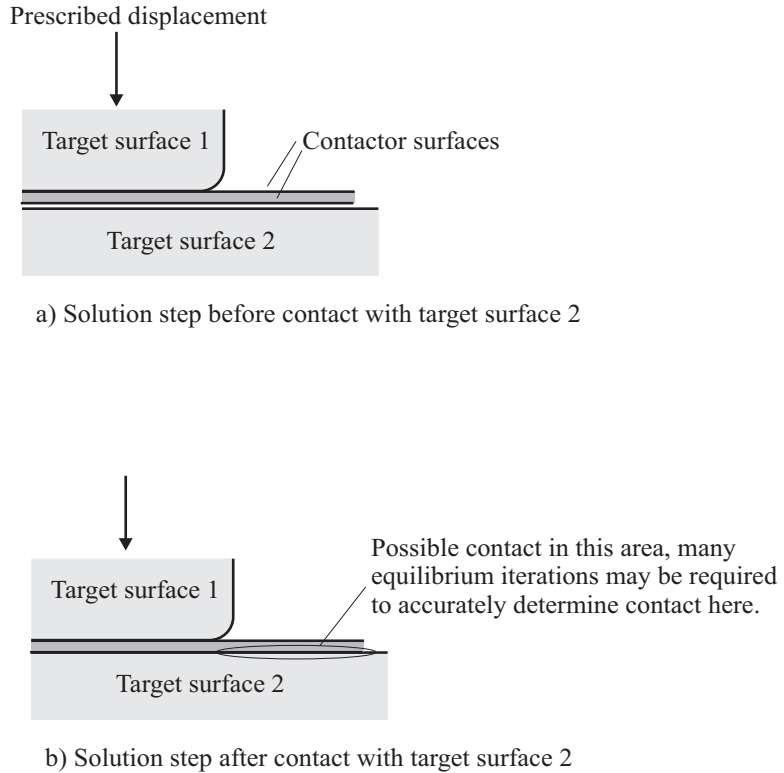


Figure 4.29-28: Establishment of contact over a large area during a solution step

- The contact search algorithm may take a relatively long time for the first iteration of the first time step. Similarly, the contact search algorithm may take a relatively long time for the first iteration of any time step in which a contact group or contact pair is born.
- As the contactor surface is refined, keeping k_n constant, the overlap and contact force will decrease at each contactor node. Hence k_n may need to be adjusted as the mesh is refined. In general, as the mesh is refined, k_n can be decreased in order to keep the overlap reasonable.

- In statics and implicit dynamics, convergence may become difficult when contactor nodes that were not in contact with the target suddenly interact with the target. An example is illustrated in Fig. 4.29-29. Eventually the contactor nodes on the right will come into contact with the target, and convergence may be difficult. Alternative ways to model this situation are shown in Fig. 4.27-29.

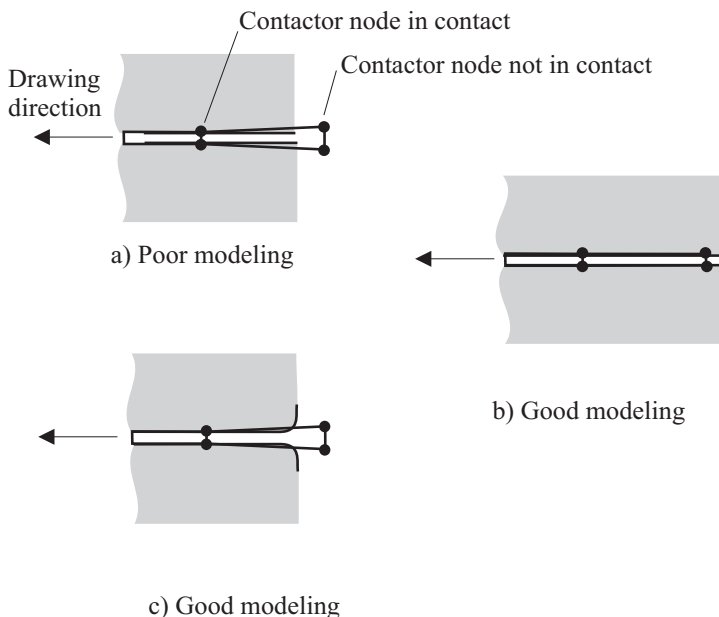


Figure 4.29-29: Contactor nodes suddenly coming into contact

Another example is shown in Fig. 4.29-30. In Fig. 4.29-30(a), the top target surface is flat, and the indicated node suddenly comes into contact with the top target surface. Convergence is very difficult, because a very small change in the position of the indicated node can cause the contact status of that node (and hence the contact force) to change abruptly. In Fig. 4.29-30(b), the top target surface has a round corner, and the indicated node gradually comes into contact with the top target surface. Convergence is easier, because a very small change of the position of the indicated node results in only a very small change in the contact force.

In metal forming analysis, this situation is frequently encountered in the modeling of the blank holder. The modeling is

easiest if the blank holder is modeled as a flat target surface. But convergence is easier if round corners are added to the blank holder wherever nodes on the blank are anticipated to contact the blank holder during drawing.

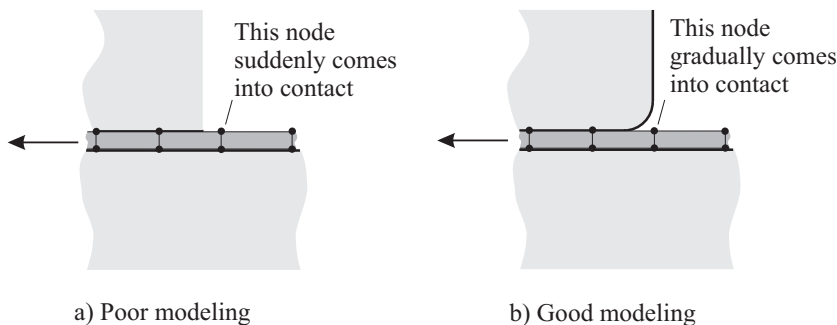


Figure 4.29-30: Additional example of contactor nodes suddenly coming into contact

4.29.7 Conversion of models set up using the rigid target algorithm of ADINA 8.3

The following hints may be useful when running models that were set up using the rigid target algorithm of ADINA 8.3 and earlier:

- The results from the V83 and current rigid target contact algorithms will usually be quite different:
 - a) In statics and implicit dynamics, the V83 rigid target algorithm only determines the state of contact at iteration 0; the current rigid target algorithm determines the state of contact at every iteration.
 - b) The V83 rigid target algorithm only allows contact between a contactor node and one target surface; the current rigid target algorithm allows contact between a contactor node and more than one target surface.
 - c) In statics and implicit dynamics, the V83 rigid target algorithm can force ATS cutbacks when it detects tensile forces,

the current rigid target algorithm does not force ATS cutbacks.

d) In explicit dynamics, the basic formulation used by the V83 rigid target algorithm is quite different than the basic formulation used by the current rigid target algorithm.

- In general, in statics and implicit dynamics, the current rigid target algorithm requires more iterations than the V83 rigid target algorithm to establish contact. This is because, in the V83 rigid target algorithm, the state of contact is only determined in iteration 0.
- Once contact is established, the current rigid target algorithm can be used with much larger time steps than the V83 rigid target algorithm. The “excessive penetration” issues of the V83 rigid target algorithm do not exist in the current rigid target algorithm.
- Remember that the current rigid target algorithm can automatically determine the number of buckets (CONTACTPAIR NX, NY, NZ). So you should check the model to see if these parameters are specified, and, if they are specified, reset them to zero.
- In the V83 rigid target algorithm, the ATS method is automatically turned on for statics and implicit dynamics. However, in the current rigid target algorithm, the ATS method is not automatically turned on. You may want to explicitly turn on the ATS method.
- The drawbead formulation is quite different in the current rigid target contact algorithm.
- In explicit dynamics, the V83 rigid target algorithm does not affect the critical time step. But the current rigid target algorithm can affect the critical time step.
- In the V83 rigid target algorithm, the amount of friction is based on the incremental displacements. In the current rigid target algorithm, the amount of friction is based on the velocity (incremental displacements divided by time step). So in frictional analysis, the time step size will affect the results in the current rigid target algorithm.

This page intentionally left blank

5. Boundary conditions/applied loading/constraint equations

5.1 Introduction

The objective of this chapter is to present the various options available in ADINA for the description of boundary conditions, applied loads and constraint equations.

- As discussed in ref. KJB, Section 3.3.2, there are two classes of boundary conditions: essential boundary conditions, such as prescribed displacement (and rotation) boundary conditions, comprise the first class; and natural boundary conditions, such as applied force and moment boundary conditions, comprise the second class.
- Displacement boundary conditions include nodal degree of freedom fixities, prescribed displacements and constraint equations.
- Force and moment boundary conditions include various types of applied loading.
- All displacement and force boundary conditions can be referred to the global Cartesian system or to predefined skew systems. For more information about skew systems, see Ref KJB, Section 4.2.2.
- The externally applied load vector used in the governing equilibrium equations is established using contributions from the various applied loads (see Chapter 7).

For concentrated loads (Section 5.2) and user-supplied loads (Section 5.14), the contributions of these nodal loads are directly assembled into the externally applied load vector.

For pressure loading, distributed loading, centrifugal loading and mass proportional loading, pipe internal pressure loading, and electromagnetic loading, ADINA first calculates the corresponding consistent nodal load vectors (consistent in the sense that the principle of virtual work is used) and then assembles these load vectors into the externally applied load vector. The evaluations of the consistent nodal load vectors for the various types of loading are described in the following sections.

- The options of prescribing nodal displacements, temperatures and temperature gradients are provided to enable you to specify the time variations of these quantities throughout the solution period.

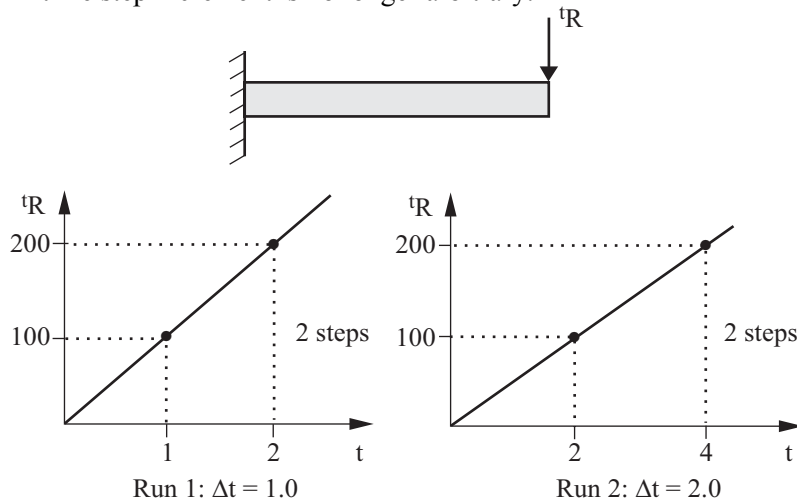
Temperatures and temperature gradients in ADINA are used in conjunction with material models which include temperature effects, see Sections 3.5 to 3.7 and 3.11.

- The definition of the time variation of the externally applied load vector for the various time steps in the solution period depends on whether automatic step incrementation is used or not.

Time variation of externally applied loads when automatic step incrementation is not used: Each applied load is associated with a time function which defines the time variation of the load throughout the solution period.

- In a static analysis in which time-dependent effects (such as creep, viscoplasticity or friction) are not included in the material models, time is a "dummy" variable which is used, via the associated time function of each applied load, to define the load intensity at a step. Thus, the time step increment directly establishes the load increments. In the example shown in Fig. 5.1-1, note that the same solution is obtained regardless of the size of the time step increment.
- In a dynamic analysis or if time-dependent effects are included in the material models in a static analysis, time is used in a similar way to define the load intensity of an applied load at a step. However, in these cases, time is a "real" variable because the time step increment is employed in the actual integration of the equations of motion in a dynamic analysis, and in the integration of the element stresses in a creep or

viscoplastic analysis. Hence, in these cases the choice of the time step increment is no longer arbitrary.



Note: identical results are obtained in Run 1 and Run 2 for a linear static analysis.

Figure 5.1-1: Example of the definition of load variation using time

Time variation of externally applied loads when automatic step incrementation is used: Two options are available:

- Using the automatic-time-stepping (ATS) procedure (see Section 7.2), the loads are defined for all times $\Delta t, 2\Delta t, \dots$ as for no automatic step incrementation. In addition, when the algorithm subdivides a time (load) step, the load vector is established by linear interpolation of the load vectors at times t and $t + \Delta t$.
- Using the load-displacement-control (LDC) procedure (see Section 7.2), the applied loads are not associated with any time function and the time variation of the loads cannot be specified by the user. The contributions from all the loads are assembled into a constant load vector denoted as the reference load vector. During the response calculation, this reference load vector is scaled proportionally using a load multiplier (in general different from one step to the next) automatically computed by the program.

- A time function is defined as a series of points $(t, f_i(t))$ in which t is time and $f_i(t)$ is the value of time function i at that time. Between two successive times, ADINA uses linear interpolation to determine the value of the time function.
- ADINA includes special time function multiplying functions that alter the time function value obtained using the $(t, f_i(t))$ input data. In the following, let $f^*(t)$ be the value of the time function obtained from the $(t, f_i(t))$ input data and let $f(t)$ be the value of the time function actually used by ADINA. Then the following multiplying functions are available:

Constant: $f(t) = f^*(t)$

Sinusoidal: $f(t) = f^*(t) \cdot \sin(\omega t + \phi)$ where ω is the angular frequency in degrees/unit time and ϕ is the phase angle in degrees.

Short circuit (type 1): $f(t) = f^*(t) \cdot \sqrt{a + b \exp(-t/\tau)}$ in which a and b are constants and τ is the time constant (unit of time).

Short circuit (type 2):

$$f(t) = f^*(t) \cdot \left(\sqrt{\frac{\mu_0}{4\pi}} \cdot \sqrt{2} \cdot I \cdot (\sin(\omega t - \phi + \alpha) + \exp(-t/\tau) \cdot \sin(\phi - \alpha)) \right)$$

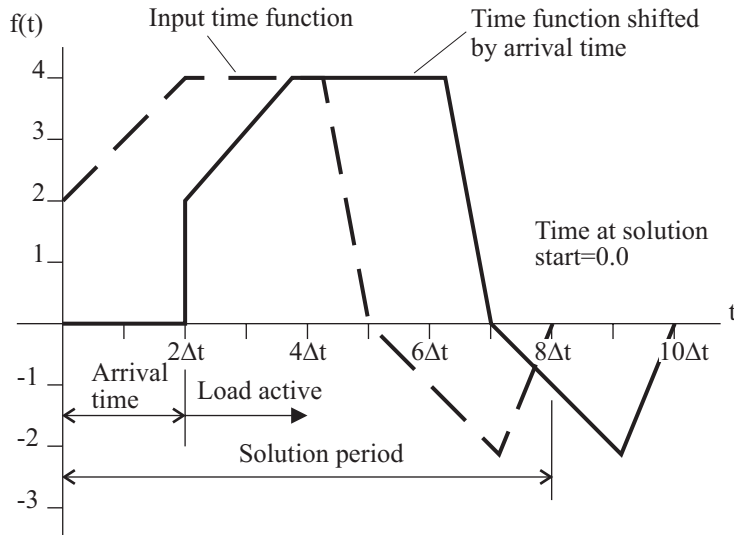
in which μ_0 is the magnetic permeability in (volt-seconds)/(meters-amperes), I is the RMS value of the short circuit current, ω is the angular frequency in degrees/unit time, ϕ is the phase angle in degrees, α is the impedance angle in degrees and τ is the time constant (unit of time).

- If time functions are used to determine the time variation of the applied loads, then the activation of the various applied loads can be delayed through the use of the arrival time option. The arrival

time option does not apply, however, to centrifugal and mass-proportional loading, see Section 5.4.

When a nonzero arrival time is used, the TFSHIFT parameter of the LOAD-OPTION command specifies whether the associated time function is shifted forward by the arrival time, see illustrations given in Fig. 5.1-2(a) and (b). By default, the time function is shifted. Note that the time function multiplier is zero for all times t smaller than the arrival time.

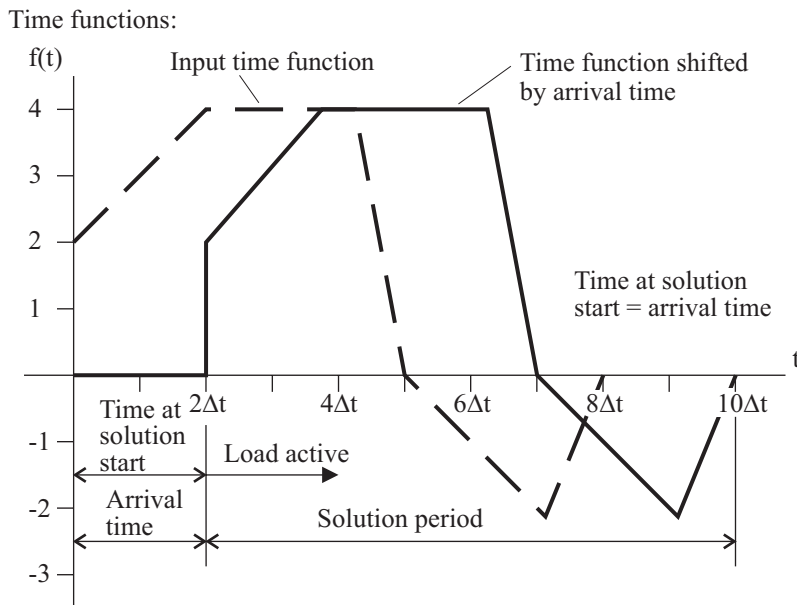
Time functions:



Resulting load values for $t_R = 10f(t)$:

Time step	1	2	3	4	5	6	7	8
$n\Delta t_R$	0	0	30	40	40	40	0	-10

Figure 5.1-2(a): Example on the use of the arrival time option



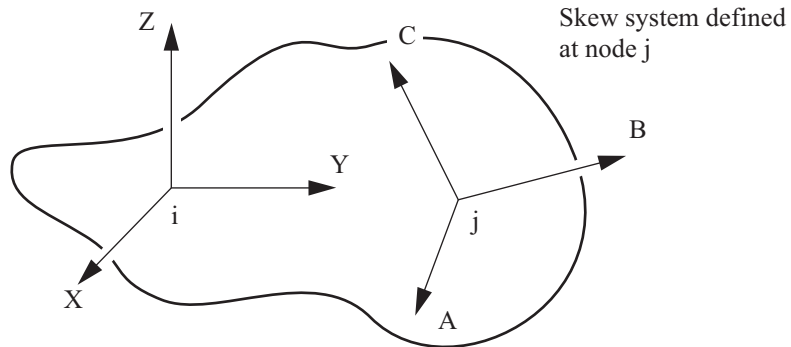
Resulting load values for $tR = 10f(t)$ and $t_0 = 2\Delta t$:

Time step	1	2	3	4	5	6	7	8
$t_0 + n\Delta t R$	30	40	40	40	0	-10	-20	0

Figure 5.1-2 (b): (continued)

5.2 Concentrated loads

- Concentrated loads are nodal point forces or moments which are applied at the specified nodes and act in the specified degrees of freedom.
- The direction in which a concentrated load acts depends on the coordinate system(s) assigned to the node of load application and upon whether the concentrated load is assigned to translational or rotational degrees of freedom; see Fig. 5.2-1.



Node	Degree of freedom (direction)	Corresponding load component
i	1	X force
	2	Y force
	3	Z force
	4	X moment
	5	Y moment
	6	Z moment
j	1	A force
	2	B force
	3	C force
	4	A moment
	5	B moment
	6	C moment

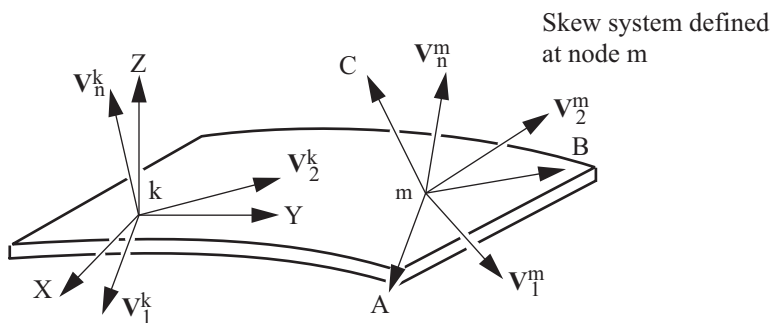
(a) Case of nodes with six degrees of freedom

Figure 5.2-1: Reference systems for application of concentrated loads

Translational degrees of freedom:

- If no skew system is defined at the node, then the concentrated loads applied to these degrees of freedom correspond to nodal forces acting in the global X, Y, and Z directions.

- If a skew system (A, B, C) is assigned at the node, then the concentrated loads applied to these degrees of freedom correspond to nodal forces acting in the A, B and C directions.



Node	Degree of freedom (direction)	Corresponding load component
k	1	X force
	2	Y force
	3	Z force
	4	V_1 moment
	5	V_2 moment
	6(*)	V_n moment
m	1	A force
	2	B force
	3	C force
	4	V_1 moment
	5	V_2 moment
	6(*)	V_n moment

(*) Note that this degree of freedom is automatically deleted by the program and moments applied at this degree of freedom are ignored.

(b) Case of nodes with five degrees of freedom

Figure 5.2-1: (continued)

Rotational degrees of freedom:

- ▶ In this case a distinction must be made between shell midsurface nodes using five degrees of freedom and the shell midsurface nodes using six degrees of freedom.
- ▶ If six degrees of freedom are used, and if no skew system is defined at the node, the concentrated loads applied to these degrees of freedom correspond to nodal moments acting about the global X, Y, and Z directions. If a skew system (A, B, C) is assigned at the node, then the concentrated loads correspond to nodal moments acting about the A, B, and C directions.
- ▶ If five degrees of freedom are used, the concentrated loads applied to these degrees of freedom correspond to nodal moments acting about the V_1 , V_2 and V_n directions of the midsurface system described in Section 2.7.
- Concentrated loads can also be specified as follower loads. In this case the direction in which a load (force or moment) acts is given by the vector pointing from an auxiliary node to the node of load application. This direction is updated during the response solution by evaluating the most current coordinates of both the node of the load application and the auxiliary node.

Note that follower moments are not allowed at shell midsurface nodes using five degrees of freedom.
- Follower loads should only be applied in a large displacement analysis. Equilibrium iterations (see Chapter 7) should, in general, be performed if follower loads are present.
- The direction of a follower load can be established with respect to a one-dimensional structure (such as a beam structure) using nonlinear rigid links (see Section 5.15.2). An example is given in Fig. 5.2-2.

5.3 Pressure and distributed loading

- Pressure loading can be applied to the surfaces of the following types of elements (see Fig. 5.3-1): **2-D solid**, **2-D fluid**, **3-D solid**, **3-D fluid**, **plate/shell** and **shell**. Considering pressure loads on

2-D elements, the area over which a pressure load acts includes the element thickness.

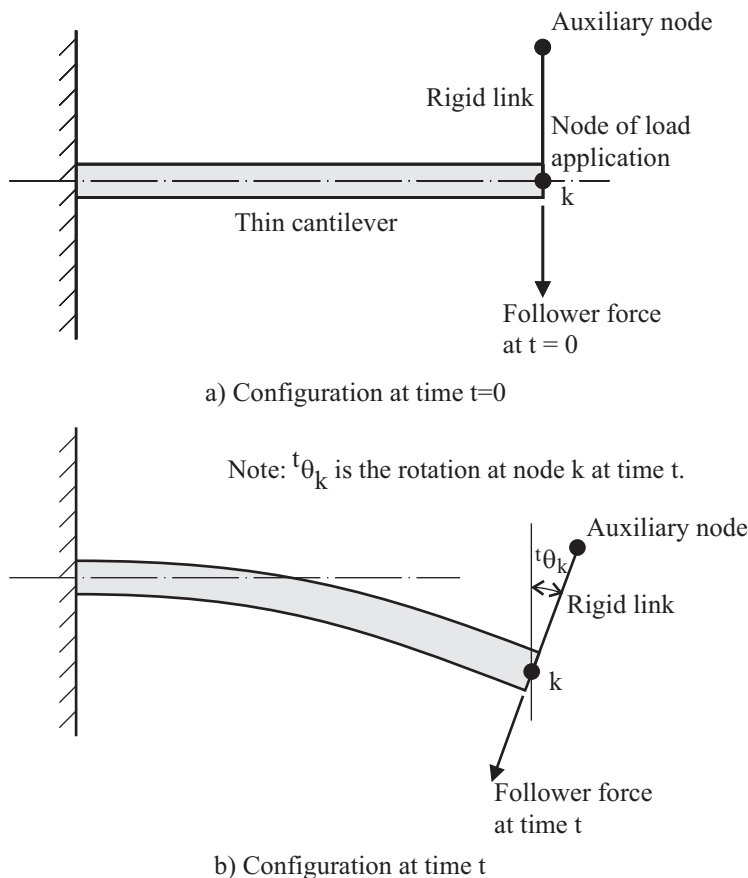


Figure 5.2-2: Example of the use of rigid link to establish the follower load direction

- Distributed loading can be applied along the longitudinal axis of the following types of elements (see Fig. 5.3-2): **beam**, **iso-beam** (including axisymmetric shells), **pipe**, **shell** (along edges).

Distributed loads do not include the element cross-section dimensions or thickness. The distributed load is applied along the neutral axis of beam/iso-beam/pipe elements, and along the midsurface line of shell elements. For example, if a shell element with original thickness of 2 undergoes uniaxial tension due to a distributed load applied to an edge of the shell element, the internal

stress is one-half the magnitude of the line load.

- For each pressure/distributed load surface specified, a consistent nodal load vector is calculated to represent the pressure/distributed loading. The formulations used in the consistent load vector calculations for the various loading surfaces are given in the following sections.

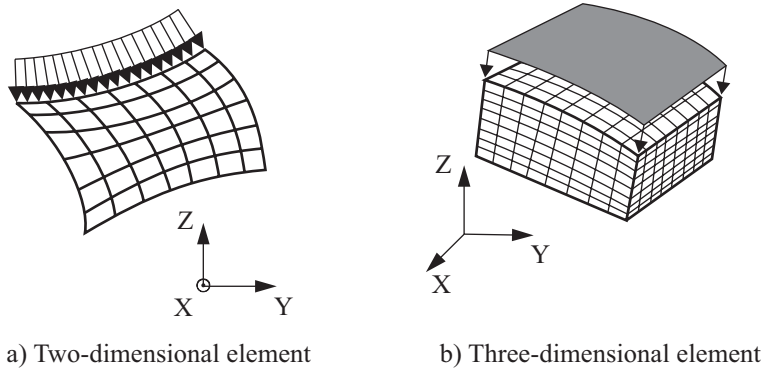


Figure 5.3-1: Examples of pressure loading

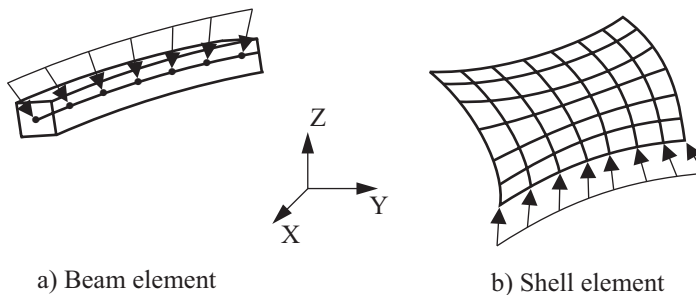


Figure 5.3-2: Examples of distributed loading

- In a large displacement analysis, the pressure/distributed loading can be specified as deformation dependent for all element types. In this case, the calculations of the consistent load vectors are based on the latest geometry and configuration of the loading surface.
- Deformation dependent loading should only be used in a large

displacement analysis. Equilibrium iterations (see Chapter 7) should in general be performed if deformation dependent loading is present.

- For pressure loads, you can optionally specify the direction of loading, as follows:
 - a) Pressure acts normal to the surface or line to which it is applied (the default).
 - b) Use x, y or z-component of a).
 - c) Pressure represents a traction in the x, y or z directions.
 - d) Pressure represents a traction tangential to the line (this option is only valid for 2-D element loadings).
 - e) Pressure represents a traction load in a user-specified direction tangent to the surface (this option is only valid for 3-D, plate/shell and shell element loadings).
- For distributed loads, you can optionally specify the direction of loading, as follows:
 - a) Distributed load acts normal to the line to which it is applied (the default).
 - b) Use x, y or z-component of a).
 - c) Distributed load represents a traction in the x, y or z directions.
 - d) Distributed load represents a traction tangential to the line.

5.3.1 Two- and three-dimensional pressure loading

- For two- and three-dimensional pressure loading, the consistent load vector consists of nodal forces acting in the translational degrees of freedom. The calculation of this load vector is given in ref. KJB, Section 4.2.1.
- Along a 2-D element side, the pressure intensity is assumed to

vary linearly (with respect to the isoparametric coordinate on the element side). Along a 3-D element face, the pressure intensity is assumed to vary bilinearly (with respect to the isoparametric coordinates on the element face).

- Deformation-dependent pressure loads that act onto the edges of plane stress elements do not take into account the change in element thickness due to in-plane deformations. For example, a plane stress element that undergoes uniaxial tension due to a deformation-dependent pressure load has internal stresses larger than the pressure load by the ratio (original thickness)/(current thickness).

Load penetration option: If a finite element "dies" due to material rupture or due to a predefined death time, the pressure loads applied to that element are transferred to the neighboring "alive" elements.

In the case when multiple pressure loads are applied to the finite element model, the total pressure applied inside a cavity created by "dead" elements is equal to the sum of the pressure loads applied to the cavity opening divided by the number of pressurized element faces at the opening (see Figure 5.3-3).

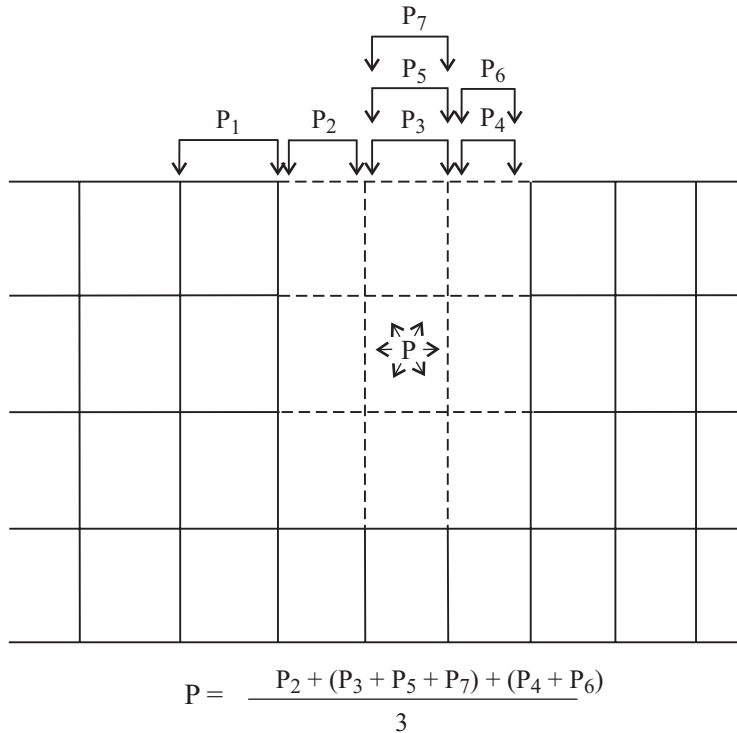


Figure 5.3-3: Pressure applied inside a domain created by "dead" elements

Figure 5.3-4 shows some examples of situations arising when the load penetration option is activated.

Load penetration can be selectively activated for designated element groups only (thus reducing the computational time necessary to perform the analysis).

In the case that a part-through crack or void is being created during the finite element analysis, load penetration should be used with caution and judgement. You must make sure that the external pressure loading is consistent at all times of the analysis with the physics of the problem solved.

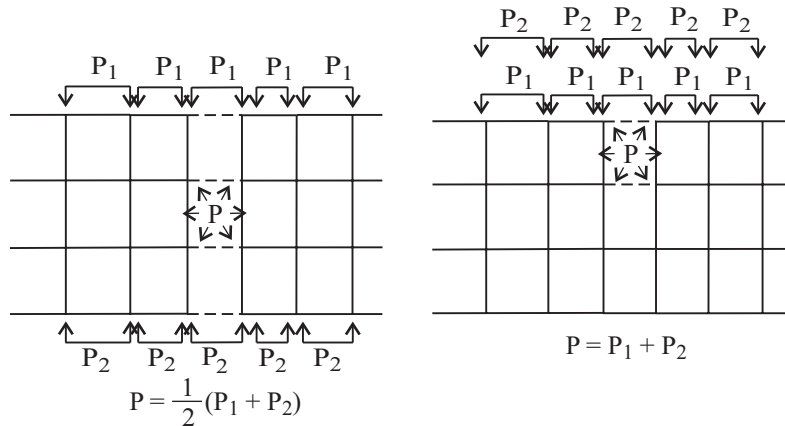


Figure 5.3-4: Some of the situations arising with load penetration

Load penetration cannot be used with deformation independent pressure loading: pressure loading must be deformation dependent.

Note that load penetration is not applicable in case of dynamic analysis using explicit time integration. Load penetration is also not applicable when the automatic LDC method is used.

5.3.2 Hermitian (2-node) beam distributed loading

- The distributed loading on a beam element is represented by the equivalent concentrated forces and moments acting at the beam nodes, see Fig. 5.3-5. These equivalent forces and moments are equal in magnitude to the fixed-end forces and moments but act in the opposite directions (the fixed-end forces and moments are the reaction forces and moments at the nodes when the beam element is subjected to the distributed load and the nodal displacements and rotations are enforced to be zero). The consistent load vector for beam distributed loading consists of these fixed-end forces and moments.

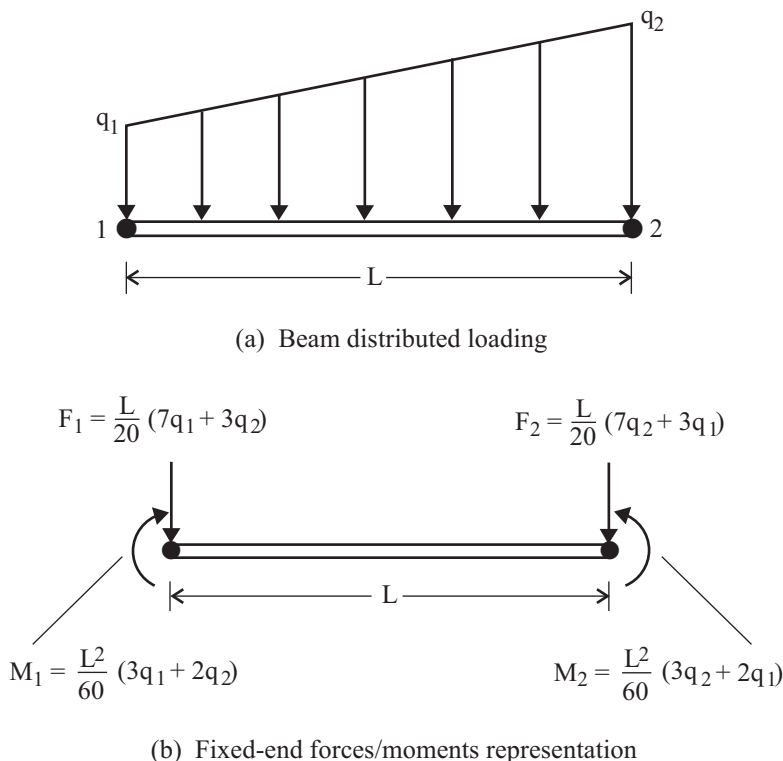


Figure 5.3-5: Representation of beam distributed loading

- Displacements and stresses in the model are calculated by representing the actual distributed loading using the consistent load vector defined above. Hence, the calculated solution corresponds only to these equivalent concentrated nodal forces and moments, and may not correspond entirely to beam theory results taking account of the distributed loading more accurately. However, an option is available to correct the force and moment results in linear static analysis to obtain the beam theory solution. This option is selected by the command MASTER FEFCORR = YES. For more information on this fixed-end force correction, see Section 2.4.5.7.

5.3.3 Iso-beam and pipe distributed loading

ref. KJB
Section 4.2.1

- The distributed loading on an iso-beam or pipe element surface is represented by concentrated forces acting at all nodes on the element longitudinal axis. These forces are calculated using the

interpolation functions for the nodal displacements as given in ref. KJB.

Since the nodal translations and rotations are interpolated independently for the iso-beam and pipe elements, the consistent load vector for the distributed loading consists of nodal forces only (no moments).

- For iso-beam and pipe elements subjected to distributed loading, the element nodal forces/moments and stresses calculated in ADINA are due to the consistent nodal loads.

5.3.4 Plate/shell pressure loading

- The pressure loading on a plate/shell element surface is represented by concentrated forces acting at the three corner nodes. These forces are calculated assuming a linear interpolation of the transverse displacements between nodes and hence this consistent load representation of the pressure loading is only approximate.

5.3.5 Shell loading

- The pressure loading on a shell element surface is represented by concentrated forces acting at the nodes on the element surface. These forces are calculated using the interpolation functions for the nodal displacements as given in ref. KJB, Section 4.2.1.

Since the nodal translations and rotations are interpolated independently for the shell element, the consistent load vector for the distributed loading consists of nodal forces only (no moments).

- Shell elements can also be subjected to a distributed line loading acting on the edge of the elements (see Fig. 5.3-2). In this case an auxiliary node needs to be used to define the plane in which the load is acting.

Deformation-dependent line loads that act onto the edges of shell elements do not take into account the change in element thickness due to in-plane deformations.

5.3.6 Contact surface modification of prescribed loads feature

- In certain problems, it is desirable for contact surfaces to control the prescribed loads acting on those surfaces. For example, there might be a prescribed pressure acting on a contact surface. It might be desirable for this prescribed pressure to be active only when the contact surface is not in contact, and for this prescribed pressure to be inactive when the contact surface is in contact (Fig. 5.3-6).

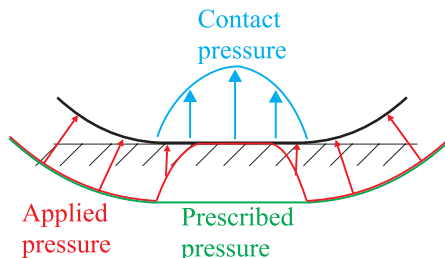


Figure 5.3-6: Concepts used in contact surface modification of prescribed loads feature

The "contact surface modification of prescribed loads" feature provides this control. This feature is off by default. The feature can be turned on using the `PRESSURE-CONTROL` parameter of the `CONTACT-CONTROL` command.

`PRESSURE-CONTROL=YES` activates the feature. `PRESSURE-CONTROL=V89` activates the feature using the version 8.9 sealing criterion (see below), but this option is only provided for backwards compatibility with version 8.9, and should not be used otherwise.

- Contact surfaces can modify the following prescribed loads: pressure loads acting on 2D, 3D, plate and shell elements. These loads must be deformation-dependent.
- The contact surfaces that can modify prescribed loads are: implicit constraint function with "new" contact surfaces, either 2D or 3D contact, either single or double-sided contact. Only contactor contact surfaces can modify prescribed loads.

- The actual applied pressure at a corner node is equal to the prescribed pressure (including the time function multiplier) multiplied by a prescribed load factor. The prescribed pressure is calculated from the pressure loads acting on 2D, 3D, plate and shell elements, and again these loads must be deformation-dependent. The load factor calculation is discussed in more detail below.

In particular, the fluid pressure from an FSI analysis does not contribute to the prescribed pressure.

- The prescribed pressure definition has the following restrictions:

a) The prescribed pressure must be computed from a single load definition. In particular, the prescribed pressure cannot be applied in two separate load definitions with two time functions.

b) The prescribed pressure must be continuous between all element faces attached to each node. However it is allowed for an element face to have a pressure load and the adjacent face to have no pressure load.

The program does not check to see if these restrictions are satisfied. But if these restrictions are not satisfied, the program will not compute the adjusted pressure correctly (see below).

- The prescribed load factor is based on the contact pressure at the node, the prescribed pressure at the node, and also on three input values: the critical pressure, the penetration time and the evacuation time. These values are input using parameters CRITICAL-PRESSURE, TIME-PENETRATION, TIME-EVACUATION of the CONTACT-CONTROL command. The default values of these parameters are zero.

Sealing and unsealing criteria

- In the evaluation of the prescribed load factor, it is useful to consider the following quantity:

$$\text{Adjusted pressure} = (\text{contact pressure}) - (\text{unapplied pressure})$$

in which

Unapplied pressure = (prescribed pressure) — (applied pressure)

In terms of the prescribed load factor,

Applied pressure = (prescribed load factor) × (prescribed pressure)

Unapplied pressure = (1 — (prescribed load factor)) × (prescribed pressure)

- We then can define the following terms:

Unsealing: The adjusted pressure is less than the critical pressure

Sealing: The adjusted pressure is greater than the critical pressure

In order to see that these definitions make sense, consider the following cases:

1) Contact is established, so that the segment is sealed. There is a contact pressure and also a prescribed pressure, but the applied pressure is zero because the segment is sealed. Now suppose that a small virtual gap is created (like a virtual displacement). In that case, the prescribed pressure acts on the segment, reducing the contact pressure to the adjusted pressure. If the adjusted pressure is greater than the critical pressure, the virtual gap will close, reestablishing the seal. But if the adjusted pressure is less than the critical pressure, then the gap will not close and the seal will begin unsealing.

2) Contact pressure less than the critical pressure, so that the segment is leaking. Both the prescribed and applied pressures are nonzero, thus the unapplied pressure is zero and the adjusted pressure is equal to the contact pressure. Now suppose that in the next step the contact pressure (and also the adjusted pressure) becomes greater than the critical pressure. The segment will begin to seal.

So the adjusted pressure is simply the contact pressure for the case

in which, for whatever reason, the segment hypothetically starts to leak.

If $\text{PRESSURE-CONTROL}=\text{V89}$, then the adjusted pressure is defined to be the contact pressure. The transition from leaking to sealing is the same as that given above. But the transition from sealing to leaking occurs only when the contact pressure itself becomes less than the critical pressure.

Evaluation of prescribed load factor

At the beginning of the first time step, the prescribed load factor is 1.0.

At the beginning of the remaining time steps, the prescribed load factor is increased or decreased according to:

Adjusted pressure less than critical pressure. The segment is unsealing. The prescribed load factor is increased by the value $(\Delta t)/(\text{penetration time})$, with a maximum value of prescribed load factor equal to 1. If the penetration time is equal to zero, the prescribed load factor is increased to 1.

Adjusted pressure greater than critical pressure. The segment is sealing. The prescribed load factor is decreased by the value $(\Delta t)/(\text{evacuation time})$, with a minimum value of prescribed load factor equal to 0. If the evacuation time is equal to zero, the prescribed load factor is decreased to 0.

Example

Figure 5.3-7 schematically shows the unsealing and sealing processes. For simplicity, only one contact segment is shown, and the critical pressure is assumed to be zero. The penetration and evacuation times are greater than zero, so that the unsealing and sealing take place over several time steps. Also for simplicity, the deformations in the remainder of the model are held fixed during the unsealing and sealing (so that the total pressure acting on the contact segment remains unchanged during the unsealing and sealing).

In Fig 5.3-7(a), the contact segment is in contact with no

prescribed pressure. In Figs 5.3-7(b) to (e), the prescribed pressure is increased. Only when the prescribed pressure equals the contact pressure (Fig 5.3-7(c)) does the applied pressure become nonzero. As the applied pressure increases, the contact pressure decreases (so that the total pressure acting on the contact segment is unchanged). When the applied pressure equals the prescribed pressure (Fig 5.3-7(e)), the contact pressure is zero and the seal is leaking.

In Figs 5.3-7(f), contact is reestablished (perhaps due to deformations acting elsewhere in the model). The contact pressure is very small, but greater than zero. The applied pressure starts to decrease. As the applied pressure decreases, the contact pressure increases further (again, so that the total pressure on the contact segment is unchanged). Eventually the applied pressure drops to zero and the seal is sealed.

Notice that the adjusted pressure is zero in Figs 5.3-7(c) to (e), and also in Figs 5.3-7(f) to (h).

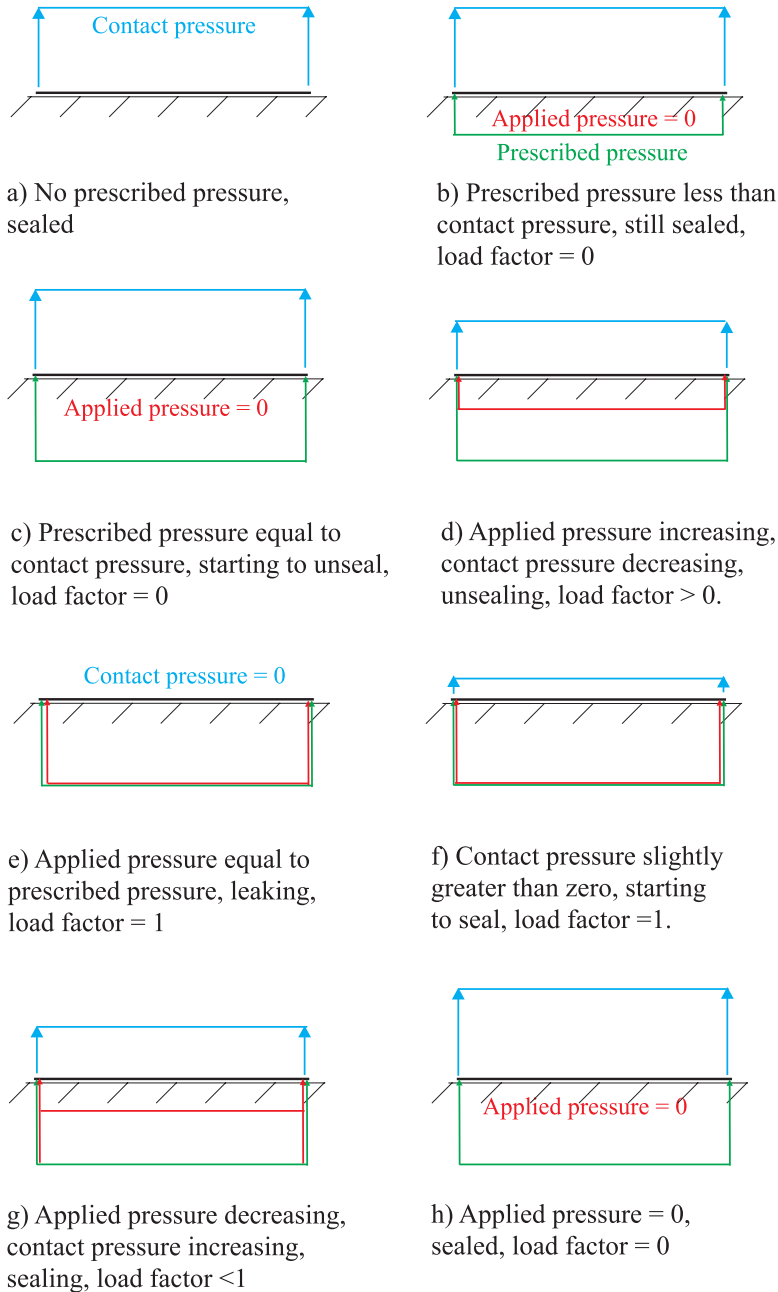


Figure 5.3-7: Schematic example showing unsealing and sealing

Hints

- The prescribed load factor is updated at the beginning of the time step, and is unchanged throughout the time step. Therefore small time steps should be used in the analysis.
- It is allowed to set the critical pressure to zero.
- It is allowed to change the critical contact pressure, penetration time and evacuation time in a restart analysis.
- The prescribed load factors can be listed using the ADINA-PLOT variable `PRESCRIBED_LOAD_FACTOR`. The prescribed load factors are used in the plots of prescribed pressures.
- The following points are important to remember:
 - a) Contact pressure is force/unit length for 2D planar contact and is force/unit area for 2D axisymmetric and 3D contact.
 - b) The prescribed load factor is only evaluated at corner nodes. The applied pressure at a mid-side node is interpolated from the applied pressures at the corner nodes, regardless of the value of the contact pressure at the mid-side node.
 - c) The contact pressure is the normal part of the contact traction vector.
 - d) Contact tractions are computed in an approximate way from the contact forces.

5.4 Centrifugal, rotational and mass proportional loading

5.4.1 Overview

- Centrifugal, rotational, and mass proportional loading can be used to model the effect of inertia forces that arise from

accelerations to which the structure is subjected:

- ▶ Centrifugal loading is used for structures that rotate about one axis of revolution.
 - ▶ Rotational loading is used for structures that rotate about two axes of revolution.
 - ▶ Mass proportional loading is used for structures that accelerate only in the translational degrees of freedom, and is commonly used to model gravity loading and ground excitation.
-
- Centrifugal loading can be applied to the whole model, to individual element groups, or to element sets. Rotational loading and mass-proportional loading can only be applied to the whole model, or to individual element groups.
 - It is not allowed to have more than one centrifugal load or rotational load applied to the whole model. However, it is allowed to have more than one centrifugal load or rotational load applied to part of a model. For example, one centrifugal load can be applied to the whole model, and another centrifugal load can be applied to an individual element group.
 - When more than one centrifugal load or rotational load is applied to part of the model, each load acts independently. This means that certain inertia forces due to the coupling of the velocities (e.g. gyroscopic forces) are not accounted for in the analysis. In certain problems, the coupling terms are significant. Therefore, in general, it is recommended to only apply one centrifugal load or rotational load to any part of the model. If a part rotates about two axes, rotational loading should be applied.
 - The time function has different meanings in centrifugal and rotational loading. In centrifugal loading, the time function describes the time variation of the inertia loads; whereby, in rotational loading, the time function describes the time variation of the angular velocities and the angular accelerations.
 - The mass matrix used in the calculation of centrifugal, rotational, and mass proportional loading can be lumped or

consistent. Note that the computational effort involved and memory required is less when a lumped mass matrix is used, but the resulting solution is, in general, less accurate.

- Centrifugal and mass proportional loading can be used in static analysis, dynamic analysis, and frequency analysis. Rotational loading can only be used in static analysis and dynamic analysis.
- The angular velocities in centrifugal and rotational loading are not coupled with the nodal velocities in the dynamic response. This means that certain inertia forces due to the coupling of the velocities (e.g. gyroscopic forces) are not accounted for in dynamic analysis. It is recommended to only use centrifugal and rotational loading in static analysis.
- In nonlinear analysis, deformation-dependent centrifugal and rotational loading is automatically requested for all element types.
- In this case, the calculation of the consistent load vector is based on the current geometry, which adds additional nonlinear terms to the stiffness matrix and to the load vector. These additional nonlinear terms account for spin softening/stiffening effects.
- In frequency analysis with centrifugal loading, only the spin softening effect from the centripetal acceleration load is accounted for, as the nonlinear stiffness matrix contribution associated with this load is symmetric. The spin softening/stiffening effect from the angular acceleration load is not accounted for, because the nonlinear stiffness matrix contribution associated with this load is skew-symmetric.
- Rotational loading is not supported in frequency analysis.
- When the element birth and death option is active (see Section 11.4), the consistent load vector only contains the contributions from the elements currently alive.
- The arrival time option (see the beginning of this chapter) is not applicable for centrifugal, rotational, and mass proportional loading.

- Centrifugal and rotational loading is not supported for potential-based elements.
- Centrifugal and rotational loads are only approximately computed for the large displacement beam element. This is because the centrifugal and rotational loads are calculated assuming that the beam is straight. The error in the solution is reduced as the beam element mesh is refined.

5.4.2 Centrifugal loading

- Centrifugal loading can be used to model the effect of inertia forces on a structure that rotates about one axis of revolution. The consistent load vector for centrifugal loading is computed as follows (see Fig. 5.4-1):

$${}^t\mathbf{R} = - \int \rho \left[\underbrace{({}^t\boldsymbol{\alpha} \times {}^t\mathbf{r})}_{\text{angular acceleration}} + \underbrace{{}^t\boldsymbol{\omega} \times ({}^t\boldsymbol{\omega} \times {}^t\mathbf{r})}_{\text{centripetal acceleration}} \right] dV \quad (5.4-1)$$

where ${}^t\boldsymbol{\alpha} = \text{ALPHA} \cdot \text{FACTOR} \cdot f(t)$ is the angular acceleration, ${}^t\boldsymbol{\omega} = \left(\text{OMEGA} \sqrt{\text{FACTOR} \cdot f(t)} \right) \mathbf{n}$ is the angular velocity, ${}^t\mathbf{r}$ is the radial vector from the axis of rotation to the node, ALPHA, OMEGA and FACTOR are constants specified in the load definition, ρ is the density, $f(t)$ is the time function and \mathbf{n} is the unit vector parallel to the axis of rotation.

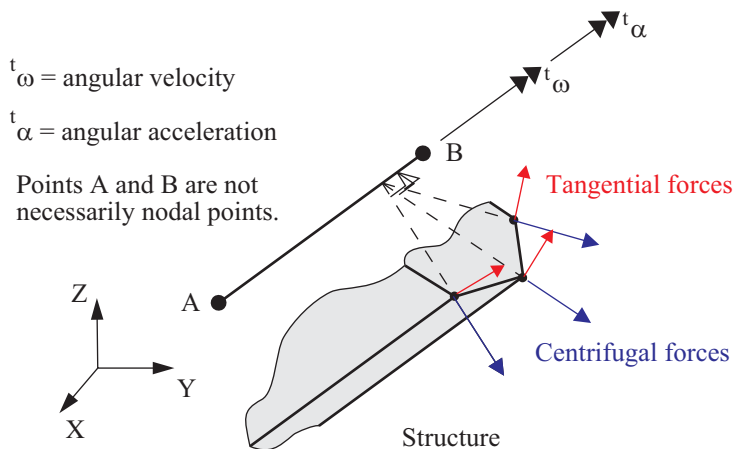


Figure 5.4-1: Convention used for centrifugal loading

Note that the time function associated with the centrifugal loading directly determines the time variation of the inertia forces and not the angular velocity.

Note that the consistent load vector consists of nodal point forces applied to all the nodes of the finite element model. But loads applied to fixed degrees of freedom do not have any effects in the solution.

- In nonlinear analysis, deformation-dependent centrifugal loading is automatically requested. In this case, the calculation of the consistent load vector is based on the current geometry, which adds additional nonlinear terms to the stiffness matrix and to the load vector. These additional nonlinear contributions are described in the following. Consider the equilibrium of the finite element system at time $t + \Delta t$, iteration i . Then

$${}^{t+\Delta t}\mathbf{M} {}^{t+\Delta t}\ddot{\mathbf{U}}^{(i)} + {}^{t+\Delta t}\mathbf{K}^{(i-1)} \Delta\mathbf{U}^{(i)} = {}^{t+\Delta t}\mathbf{R}^{(i)} - {}^{t+\Delta t}\mathbf{F}^{(i-1)} \quad (5.4-2)$$

where \mathbf{M} = mass matrix, $\mathbf{K} = \text{stiffness matrix} = (\mathbf{K}_L + \mathbf{K}_{NL_0})$, $\ddot{\mathbf{U}}$ = acceleration vector, $\Delta\mathbf{U}$ = incremental displacement vector, \mathbf{R} = external load vector, \mathbf{F} = nodal point forces corresponding to stresses, (i) = iteration i , and $t + \Delta t$ = time $t + \Delta t$.

The complete expression of the centripetal acceleration

contribution used by ADINA, including all nonlinear effects is

$${}^{t+\Delta t}\mathbf{R}^{(i)} = - \int \rho \left(\boldsymbol{\omega} \times \left(\boldsymbol{\omega} \times \left({}^0\mathbf{r} + {}^{t+\Delta t}\mathbf{U}^{(i-1)} + \Delta\mathbf{U}^{(i)} \right) \right) \right) dV \quad (5.4-3)$$

where ${}^0\mathbf{r}$ = vector of initial nodal distances to rotation axis and $\boldsymbol{\omega}$ = angular velocity vector. From the expression (5.4-3), it can be seen that a deformation-dependent load is present, given by

$${}^{t+\Delta t}\mathbf{K}_{NL_1}^{(i-1)} \Delta\mathbf{U}^{(i)} = \int \rho \left(\boldsymbol{\omega} \times \left(\boldsymbol{\omega} \times \Delta\mathbf{U}^{(i)} \right) \right) dV$$

and that an additional nonlinear contribution \mathbf{K}_{NL_1} to the stiffness matrix is present, given by

$${}^{t+\Delta t}\mathbf{R}_{NL}^{(i)} = - \int \rho \left(\boldsymbol{\omega} \times \left(\boldsymbol{\omega} \times {}^{t+\Delta t}\mathbf{U}^{(i-1)} \right) \right) dV$$

The additional stiffness contribution \mathbf{K}_{NL_1} is symmetric. Since the tangent stiffness matrix includes this term, spin softening effects from the centripetal acceleration contribution is accounted for in frequency analysis.

The complete expression of the angular acceleration load contribution used by ADINA, including all nonlinear effects is

$${}^{t+\Delta t}\mathbf{R}^{(i)} = - \int \rho \left(\boldsymbol{\alpha} \times \left({}^0\mathbf{r} + {}^{t+\Delta t}\mathbf{U}^{(i-1)} \right) \right) dV$$

In this case, a deformation-dependent load is present, given by

$${}^{t+\Delta t}\mathbf{R}_{NL}^{(i)} = - \int \rho \left(\boldsymbol{\alpha} \times {}^{t+\Delta t}\mathbf{U}^{(i-1)} \right) dV$$

However, ADINA does not include the nonlinear stiffness matrix contribution from the angular acceleration load. The stiffness matrix associated with this load is skew-symmetric due to the single cross-product term. Since the tangent stiffness matrix does

not include this term, the spin stiffening/softening effect from the angular acceleration load is not accounted for in frequency analysis.

- The correction to the stiffness matrix and the correction to the loading are made when a deformation dependent loading is requested.

5.4.3 Rotational loading

- Rotational loading can be used to model the effect of inertia forces on a structure that rotates about two axes of revolution at a particular “snapshot” in time. The feature accounts for centripetal, Coriolis, and angular accelerations, and nonlinear spin softening/stiffening effects.
- When a structure rotates about two axes, the problem is not steady-state in a Lagrangian sense. In this case, the rotational inertia forces acting on a material particle change as a function of time, even if the angular velocity is constant. Hence, rotational loading can only be used to model the effect of the inertia forces at a particular “snapshot” in time.
- A reference frame is used to describe the motion of the structure. The reference frame rotates with angular velocity $\boldsymbol{\Omega}_{frame}$ and angular acceleration $\boldsymbol{\alpha}_{frame}$. The structure rotates with angular velocity $\boldsymbol{\omega}_{rel}$ and angular acceleration $\boldsymbol{\alpha}_{rel}$ relative to this reference frame (see Fig. 5.4-2).
- The position and direction of $\boldsymbol{\omega}_{rel}$ and $\boldsymbol{\alpha}_{rel}$ remains fixed with respect to the reference frame. The total angular velocity of the structure is $\boldsymbol{\omega}_{tot} = \boldsymbol{\Omega}_{frame} + \boldsymbol{\omega}_{rel}$, and the total angular acceleration is $\boldsymbol{\alpha}_{tot} = \boldsymbol{\alpha}_{frame} + \boldsymbol{\alpha}_{rel}$.

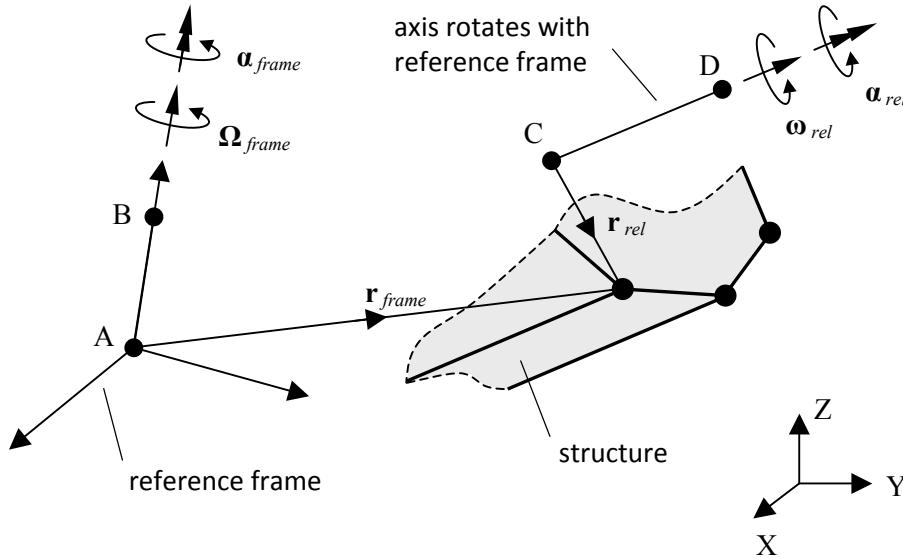


Figure 5.4-2: Convention used for rotational loading

- In Fig. 5.4-2, points A, B, C, and D are not necessarily nodal points. These points remain fixed in the reference frame throughout the analysis.
- The consistent load vector for rotational loading is computed as follows (see Fig. 5.4-2):

$$\begin{aligned}
 {}^t\mathbf{R} = \int -\rho \left\{ \underbrace{\left({}^t\boldsymbol{\alpha}_{frame} \times {}^t\mathbf{r}_{frame} + {}^t\boldsymbol{\alpha}_{rel} \times {}^t\mathbf{r}_{rel} \right)}_{\text{Angular Acceleration}} \right. \\
 + \underbrace{\left({}^t\boldsymbol{\Omega}_{frame} \times \left({}^t\boldsymbol{\Omega}_{frame} \times {}^t\mathbf{r}_{frame} \right) + {}^t\boldsymbol{\omega}_{rel} \times \left({}^t\boldsymbol{\omega}_{rel} \times {}^t\mathbf{r}_{rel} \right) \right)}_{\text{Centripetal Acceleration}} \\
 \left. + \underbrace{2 \, {}^t\boldsymbol{\Omega}_{frame} \times \left({}^t\boldsymbol{\omega}_{rel} \times {}^t\mathbf{r}_{rel} \right)}_{\text{Coriolis Acceleration}} \right\} dV
 \end{aligned} \quad (5.4-4)$$

where ρ is the density.

- The time functions describe the time variation of the angular velocities and angular accelerations (and not of the inertia forces). For example, ${}^t\boldsymbol{\Omega}_{frame} = \text{OMEGA } f(t)\mathbf{n}$, where OMEGA is a constant specified in the REF-FRAME command, $f(t)$ is the time function, and \mathbf{n} is the unit vector corresponding to the axis of revolution of the reference frame. Independent time functions are used for ${}^t\boldsymbol{\Omega}_{frame}$, ${}^t\boldsymbol{\alpha}_{frame}$, ${}^t\boldsymbol{\omega}_{rel}$, and ${}^t\boldsymbol{\alpha}_{rel}$.
- Rotational loading accounts for angular acceleration, centripetal acceleration, and Coriolis accelerations. The Coriolis accelerations give rise to gyroscopic effects.
- If the reference frame is stationary, such that $\boldsymbol{\Omega}_{frame} = \boldsymbol{\alpha}_{frame} = \mathbf{0}$, the rotational load feature applies the same consistent load vector as the centrifugal load feature.
- In nonlinear analysis, deformation-dependent rotational loading is automatically requested. In this case, the calculation of the consistent load vector in eq. (5.4-4) is based on the current geometry, which adds additional nonlinear terms to the stiffness matrix and to the load vector. The additional nonlinear terms corresponding to the centripetal acceleration and angular acceleration forces are discussed in Section 5.4.2. The additional nonlinear terms corresponding to the Coriolis acceleration forces can be obtained in a similar manner. A deformation-dependent load is present for all load contributions, however, only the stiffness matrix contribution from the centripetal acceleration load is present. ADINA does not include the stiffness matrix contribution from the angular acceleration load and the Coriolis acceleration load, as the stiffness matrix associated with these loads is, in general, not symmetric.
- The additional nonlinear terms for the centripetal acceleration forces reduce the apparent stiffness of the structure. This is known as “spin softening”. However, the additional nonlinear terms for the Coriolis acceleration can increase or reduce the apparent stiffness of the structure, depending on $\boldsymbol{\Omega}_{frame}$ and $\boldsymbol{\omega}_{rel}$.

Physically, as the structure deforms under the centrifugal forces, the radial distance increases which increases the centrifugal force. However, as the structure deforms under the Coriolis forces, the deformation may increase or reduce the Coriolis force.

5.4.4 Mass proportional loading

- The consistent load vector for mass proportional loading in direction i is computed using the mass matrix of the entire finite element system and the specified accelerations (only in the translational degrees of freedom), as follows:

$${}^t\mathbf{R}_i = {}^t\mathbf{M} \mathbf{d}_i {}^t a_i \quad (5.4-5)$$

where \mathbf{d}_i is a direction vector with "1" in the portions of the translational degrees of freedom acting into the direction i and "0" in the other portions.

The values of the accelerations are specified as follows:

$${}^t a_i = \text{MAGNITUDE} \cdot f(t) \cdot A_i, \quad i = x, y, z$$

where MAGNITUDE and A_i are specified in the load definition and $f(t)$ is the time function.

Note that the consistent load vector includes nodal point forces applied to all nodes of the finite element model.

In the calculation, the mass coupling term between active and deleted degrees of freedom is included. This mass coupling can be clearly seen in the discussion of ground motion loads later in this section.

- Mass proportional loading is commonly used to model gravity loading. For gravity loading, ${}^t a_i$ is the acceleration vector due to gravity. For example, for the z coordinate in the vertical direction (increasing z corresponds to movement away from the ground), enter ${}^t a_z = -g$, where g is the (positive) acceleration due to gravity.
- Mass proportional loads can be applied to the whole model (model-based) or to a portion of a model (group-based). Any number of element groups can be included in the group-based mass

proportional load calculations. Nodal forces for group-based mass proportional loads are calculated in the same way as for model-based mass proportional loads.

- Mass-proportional loads are frequently used to model ground motions. The basis for using mass-proportional loads in modeling ground motions is given briefly now. The equations of motion for linear dynamics, not including damping but including ground motions, are

$$\begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{12}^T & \mathbf{M}_{22} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{x}}_1 \\ \ddot{\mathbf{x}}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{12}^T & \mathbf{K}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{R}_2 \end{bmatrix} \quad (5.4-6)$$

where \mathbf{x}_1 is the vector of nodal point displacements for nodes not attached to the ground and \mathbf{x}_2 is the vector of nodal point displacements for nodes attached to the ground. \mathbf{R}_1 , \mathbf{R}_2 are externally applied forces (for example, concentrated forces).

When the ground motions are the same at all nodes attached to the ground, $\mathbf{x}_1 = \mathbf{u}_1 + \mathbf{d}_{1i}x_{gi}$, $\ddot{\mathbf{x}}_1 = \ddot{\mathbf{u}}_1 + \mathbf{d}_{1i}\ddot{x}_{gi}$, $\mathbf{x}_2 = \mathbf{u}_2 + \mathbf{d}_{2i}x_{gi}$, $\ddot{\mathbf{x}}_2 = \ddot{\mathbf{u}}_2 + \mathbf{d}_{2i}\ddot{x}_{gi}$, where \mathbf{u}_1 is the vector of nodal point displacements relative to the ground for nodes not attached to the ground and \mathbf{u}_2 is the vector of nodal point displacements relative to the ground for nodes attached to the ground. Clearly, $\mathbf{u}_2 = \mathbf{0}$. Also \mathbf{d}_{1i} is the direction vector for the nodes not attached to the ground and \mathbf{d}_{2i} is the direction vector for the nodes attached to the ground, with “1” in the portions of the translational degrees of freedom acting into the direction i and “0” in the other portions.

The matrix equation of motion becomes

$$\begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{12}^T & \mathbf{M}_{22} \end{bmatrix} \left(\begin{bmatrix} \ddot{\mathbf{u}}_1 \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{d}_{1i}\ddot{x}_{gi} \\ \mathbf{d}_{2i}\ddot{x}_{gi} \end{bmatrix} \right) + \begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{12}^T & \mathbf{K}_{22} \end{bmatrix} \left(\begin{bmatrix} \mathbf{u}_1 \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{d}_{1i}x_{gi} \\ \mathbf{d}_{2i}x_{gi} \end{bmatrix} \right) = \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{R}_2 \end{bmatrix} \quad (5.4-7)$$

Now $\begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{12}^T & \mathbf{K}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{d}_{1i} \\ \mathbf{d}_{2i} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}$ since the vector $\mathbf{d}_i = \begin{bmatrix} \mathbf{d}_{1i} \\ \mathbf{d}_{2i} \end{bmatrix}$ corresponds to a rigid body motion. The matrix equation of motion becomes

$$\begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{12}^T & \mathbf{M}_{22} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{u}}_1 \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{12}^T & \mathbf{K}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{R}_2 \end{bmatrix} - \begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{12}^T & \mathbf{M}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{d}_{1i} \ddot{x}_{gi} \\ \mathbf{d}_{2i} \ddot{x}_{gi} \end{bmatrix} \quad (5.4-8)$$

and therefore the system of equations solved is

$$\mathbf{M}_{11} \ddot{\mathbf{u}}_1 + \mathbf{K}_{11} \mathbf{u}_1 = \mathbf{R}_1 - \mathbf{M}_{11} \mathbf{d}_{1i} \ddot{x}_{gi} - \mathbf{M}_{12} \mathbf{d}_{2i} \ddot{x}_{gi} \quad (5.4-9)$$

It is seen that the ground acceleration can be applied to the model as a mass-proportional load, provided that the resulting nodal point motions are interpreted as motions relative to the ground. The mass coupling term between active and deleted degrees of freedom $\mathbf{M}_{12} \mathbf{d}_{2i} \ddot{x}_{gi}$ is included.

Please note:

- ▶ To enter a positive ground acceleration \ddot{x}_{gi} , specify a negative mass-proportional load a_i .
- ▶ All fixities are relative to the ground. In other words, fixing a node attaches it to the ground.
- ▶ All prescribed displacements are relative to the ground.
- ▶ All single DOF springs are attached to the ground.

Damping can be used. However, concentrated dampers and single DOF damping spring elements are attached to the ground. Mass-proportional Rayleigh damping, acts relative to the ground motion.

Although we have illustrated the procedure only for linear dynamics, the procedure is also valid in nonlinear dynamics.

- In a nonlinear static analysis, the nonlinear response due to gravity loading alone can be obtained first and then the structure can be subjected to additional concentrated, surface pressure loading, etc. In this case, the total gravity loading is applied alone in the first N steps, see Fig. 5.4-3. The time function defines the load increments with which the gravity loading is applied. Other concentrated loads, pressure loading, etc. are applied in the load

steps by using either the arrival time option or by the use of another time function. Hence, in this case during the last (NSTEP-N) steps, the response of the structure is due to both the maximum gravity loading and any additional concentrated, pressure loading, etc.

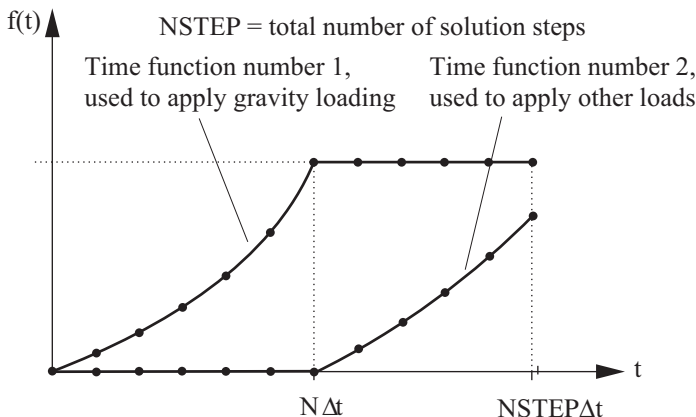


Figure 5.4-3: Use of gravity loading in nonlinear static analysis with other types of loading

5.5 Prescribed displacements, velocities & accelerations

- In some analyses, it is necessary to prescribe displacements at specified degrees of freedom. In ADINA, the following relationship can be specified.

$$u_i = f(t)$$

where f is a general time function (see Section 5.1) which prescribes the nodal point displacement u_i . Prescribed displacement conditions can be used in static and dynamic analysis.

- Nodal point translations and rotations can be prescribed. The degree of freedom for which a displacement is prescribed depends on the coordinate system assigned at the node. The convention used is exactly the same as the convention used for concentrated loads, see Section 5.2.
- Ovalization and warping modes can be prescribed at the nodes of pipe-shell elements.

- A nodal point can be "fixed" by prescribing a zero displacement component for all degrees of freedom at this node. This is, however, different from imposing a fixity boundary condition at the nodal degrees of freedom because the prescribed degrees of freedom are retained in the system matrices (i.e., equation numbers are assigned) whereas the degrees of freedom at which fixity conditions are imposed are deleted in the system matrices.
- A node which is "fixed" by having zero prescribed displacements at a step can displace due to nonzero prescribed displacements at the same degrees of freedom at another step. This technique can be used to model moving boundaries.
- In the Newmark method, the central difference method is used to calculate velocities and accelerations for nodes with prescribed degrees of freedom. Note that in the case of unloading from prescribed displacements, the program does not switch to the central difference method. Also note that there is no switch to the central difference method if the Bathe method is used.
- *Arrival time option:* When the arrival time is specified, the prescribed displacement is not applied to the node until the arrival time. Before the arrival time, the node has no load (in other words, zero prescribed force).
- *Relative prescribed displacement option:* Normally the specified prescribed displacement is directly applied to the model, so that the prescribed displacement at the node is equal to the specified prescribed displacement. However, there is a "relative prescribed displacement option" that works as follows. When the relative prescribed displacement option is active, then the prescribed displacement at the node is calculated using the specified prescribed displacement plus a preexisting displacement. The preexisting displacement is the displacement at the arrival time (if the arrival time is nonzero), or at the beginning of the current analysis (which might be a restart analysis).

The time function for the specified prescribed displacement can be constant or time-dependent.

As a special case, the specified prescribed displacement can be

zero. Then the prescribed displacement is equal to the preexisting displacement. This option is useful for “freezing” the motion of a node, so that before the arrival time, the node has no prescribed displacement, and after the arrival time, the node has a prescribed displacement equal to the displacement at the arrival time.

- *Unloading options:* Normally the prescribed displacement remains throughout the analysis. However, there are unloading options that allow the prescribed displacement to change into a prescribed force.

Time unloading: After the user-specified unloading time, the prescribed displacement changes into a prescribed force. The value of the prescribed force is calculated as a user-specified unloading force multiplied by the value of the unloading time function. If the user-specified unloading force value is zero, then the value of the prescribed force is calculated as the reaction force at the unloading time multiplied by the value of the unloading time function.

Force unloading: When the reaction force is larger than the user-specified prescribed force, the value of the prescribed force is calculated as the user-specified unloading force multiplied by the value of the unloading time function. It is an error to enter a user-specified unloading force value of zero.

Although the options are termed “unloading” options, it is not necessary for the prescribed force to actually decrease after the unloading option changes the prescribed displacement into a prescribed force.

Typical uses of the unloading options are:

- 1) Unloading of a formed part. It is then convenient to use time unloading with a zero user-specified prescribed force. The value of the unloading time function is then the ratio of the current prescribed force to the reaction value at the specified unloading time.
- 2) Application of a constant blank holding force. Either use time unloading with a nonzero user-specified unloading force, or use force unloading with a user-specified unloading force.

3) Unfreezing of a node. Before the unloading time, the displacement of the node is prescribed (usually to zero), after the unloading time, the node is free (with zero prescribed force). Use time unloading with any (zero or nonzero) user-specified unloading force, and an unloading time function which is always equal to zero.

The process by which prescribed displacements change into prescribed forces are illustrated by the following examples.

1) Time unloading with zero user-specified unloading force. Suppose that the unloading time is 2.5, the solution times are 1.0, 2.0, 3.0, 4.0 etc. The solution for time 3.0 is computed as usual with a prescribed displacement. The program then saves the reaction force at time 3.0 for further use. The solutions for times 4.0, 5.0, etc. are computed using a prescribed force calculated as the reaction force at time 3.0 multiplied by the unloading time function evaluated at times, 4.0, 5.0, etc.

2) Time unloading with nonzero user-specified unloading force. Suppose that the unloading time is 2.5, the solution times are 1.0, 2.0, 3.0, 4.0 etc. The solution for time 3.0 is computed as usual with a prescribed displacement (not with the user-specified prescribed force). The solutions for times 4.0, 5.0, etc. are computed using a prescribed force calculated as the user-specified unloading force multiplied by the unloading time function evaluated at times 4.0, 5.0, etc.

3) Force unloading. Suppose that the user-specified unloading force is 1000.0. At time 2.0, the solution is obtained using a prescribed displacement, and the reaction force is 700.0. At time 3.0, the solution is obtained using a prescribed displacement, and the reaction force is 1200.0. Since the reaction force is larger than the user-specified unloading force, the solutions for the successive times 4.0, 5.0, etc. are computed using prescribed forces as follows. The solutions for times 4.0, 5.0, etc are computed using a prescribed force calculated as the user-specified unloading force (1000.0) multiplied by the unloading time function evaluated at times 4.0, 5.0, etc.

- Prescribed velocities and accelerations can only be used in

dynamic direct time integration solutions using the following methods:

- Newmark
- Bathe method
- explicit time integration
- Prescribed velocities and accelerations can be applied in the same way as prescribed displacements, except that:
 - the unloading option cannot be used with prescribed velocities and accelerations
 - the relative prescribed option is not applicable to prescribed velocities and accelerations
- Prescribed rotations can be applied to the rotational degrees of freedom of large displacement formulation structural elements (such as Hermitian beam elements), and also to the master rotational degrees of freedom of large displacement rigid links. The formula used is now discussed.

In large displacement analysis, the increment in prescribed rotation is computed as $\Delta\theta = {}^{t+\Delta t}\theta - {}^t\theta$, where ${}^t\theta$, ${}^{t+\Delta t}\theta$ are the prescribed rotations at times t , $t + \Delta t$. The components of the increment in prescribed rotation are $\Delta\theta = (\Delta\theta_x, \Delta\theta_y, \Delta\theta_z)$.

Consider an incremental rotation with components $\Delta\theta = (\Delta\theta_x, \Delta\theta_y, \Delta\theta_z)$ applied to the master node of a large displacement rigid link with end coordinates ${}^t\mathbf{x}_m$, ${}^t\mathbf{x}_s$. ("m" = master, "s" = slave). The vector connecting these coordinates is ${}^t\mathbf{x}_{sm} = {}^t\mathbf{x}_s - {}^t\mathbf{x}_m$. When this incremental rotation is applied to the master node of the rigid link, the vector ${}^t\mathbf{x}_{sm}$ changes to

$${}^{t+\Delta t}\mathbf{x}_{sm} = \mathbf{Q}(\Delta\theta){}^t\mathbf{x}_{sm}, \text{ where } \mathbf{Q}(\Delta\theta) = \mathbf{I} + \frac{\sin \gamma}{\gamma}\mathbf{S} + \frac{1}{2}\left(\frac{\sin \frac{\gamma}{2}}{\frac{\gamma}{2}}\right)\mathbf{S}^2$$

$$\text{and } \mathbf{S} = \begin{bmatrix} 0 & -\Delta\theta_z & \Delta\theta_y \\ \Delta\theta_z & 0 & -\Delta\theta_x \\ -\Delta\theta_y & \Delta\theta_x & 0 \end{bmatrix}, \gamma = \sqrt{\Delta\theta_x^2 + \Delta\theta_y^2 + \Delta\theta_z^2}. \text{ The}$$

above formula can be found in ref KJB, p 580, exercise 6.56.

This formula gives the exact answer for the case when the incremental rotations are finite. Physically, the incremental rotations are interpreted as a vector, with the direction of the vector interpreted as the axis of rotation, and the magnitude of the vector interpreted as the amount of rotation about that axis.

The same formula is applied to the end-node triad of a beam element local node, and to the director vector of a shell element local node.

Calculation of accelerations and reactions for nodes with prescribed displacements or velocities

- In dynamic analysis, if the prescribed displacements or prescribed velocities vary nonlinearly in time, the accelerations at the prescribed DOF are very sensitive to the number of time function points used to specify the prescribed motion. The reason is that the accelerations are computed from the second time derivative of displacement, so any error in displacement is magnified by the differentiation.
- To compute the accelerations as accurately as possible there should be a time function point corresponding to every solution time, including the solution times corresponding to sub-steps in the Bathe and Noh-Bathe methods, and including the solution times corresponding to ATS cutbacks. For example, in an analysis in which the Bathe method is used, and assuming that there are no ATS cutbacks, there should be a time function point for all times $t + \gamma\Delta t$ as well as for times $t + \Delta t$.
- Sometimes it is not possible to know in advance the solution times for which ADINA evaluates the time function, due to possible ATS cutbacks during the solution. Also the solution times used by ADINA are sometimes not evenly spaced, for example in linear analysis with the Bathe method, in which $\gamma = 2 - \sqrt{2}$.

- If there is no time function point at a given solution time, ADINA uses linear interpolation in time between the two bracketing times at which there are time function points. Depending upon the actual (mathematical) shape of the time function, this linear interpolation may or may not cause inaccuracies in the accelerations. For example, if the mathematical shape of the time function is linear at each solution time, then the linear interpolation does not cause any inaccuracy. However, if the time function is varying nonlinearly, the linear interpolation can cause unwanted inaccuracies.
- The most general way to avoid the inaccuracies is to sample the (mathematical) time function using more than one time function point per solution step. For example, supposing that the solution is to be obtained at times 0.01, 0.02, etc., the time function could be sampled using a much smaller time step size, e.g. 0.001, 0.002, etc. The speed of the program solution is not affected by the number of time function points in the time function, so there is no drawback in specifying a large number of time function points.
- Reasonable guidelines are to use at least 2 time function points for each time step when using the Bathe method in nonlinear analysis (with the default $\gamma = 0.5$), and at least 20 time function points for each time step when using the Bathe method in linear analysis (since $\gamma = 2 - \sqrt{2}$ by default in linear analysis), the additional points ensure that at least one point is close to the solution time.
- The reactions at nodes with prescribed displacements or velocities can also be inaccurate, for the same reasons as described above.
- In some cases, inaccurate reactions can be detected by computing the energies for the system (see Section 11.7). When the reactions are inaccurate, the energies do not balance.
- The solutions at other nodes (nodes with free DOFs) are not affected by the above issues.

5.6 Prescribed temperatures and temperature gradients

- If temperature-dependent material models are used in the analysis, then the nodal point temperatures for all nodes must be defined at all solution steps.
- Nodal temperature gradients must be defined if shell elements or axisymmetric shell elements are present and temperature-dependent material models are used in the analysis. Nodal temperature gradients should only be assigned to shell midsurface nodes and/or axisymmetric shell nodes. The direction of the temperature gradient at a node is the same as the direction of the normal vector (or the midsurface director vector) at the node, see Section 2.7.1.
- Temperature data can be obtained from an ADINA-T analysis in which the same finite element mesh is used and thus the number of nodal points, the number of elements, etc., are the same as in the ADINA analysis. In this case ADINA-T directly creates a file containing the temperatures. This file is written as an unformatted FORTRAN file with the following line:

```
WRITE (56) TIME, (TEMP(I), I=1, NUMNP)
```

See Chapter 8 of the ADINA-T Theory and Modeling Guide for more information.

- Temperature data can also be obtained from an ADINA-T or ADINA-F analysis in which a different finite element mesh is used. In this case, ADINA-T and ADINA-F create mapping files, which are then read by the AUI in order to create temperature files for the ADINA analysis.
- Also, the nodal temperatures for all solution steps can be defined via direct input, in which case the time variations of the temperatures are specified by time functions. In the following, we describe how this is done using the AUI.
The formulas used to determine the prescribed nodal temperature at node i are

${}^t\theta_i = \text{TLOAD}$, if no temperature is prescribed at the node

and

$${}^t\theta_i = \frac{1}{N} \sum_{j=1}^N f_j(t) F_i^j$$

In these formulas, TLOAD is the overall prescribed temperature (also referred to as the default prescribed temperature or the reference prescribed temperature), N = number of temperature load applications at the node, f_j is the time function associated with load application j (not necessarily time function number j) and F_i^j is the multiplying factor associated with load application j .

- If the arrival time option is used in the nodal temperature definition, then the time function value for time $t \leq \text{ARTM}$ is zero and hence a thermal strain will result if the initial temperature is nonzero. In order to obtain a zero thermal strain prior to the arrival time, multiple temperature data sets can be assigned to the same nodal points. For example, when temperatures are prescribed directly to the nodes (not through geometry), this can be done as shown in Fig. 5.6-1.
- If temperatures are obtained both from a separate file and from direct input, the final temperatures used in the analysis are obtained by adding the temperatures from the file and the direct input.
- Temperature gradients at nodes can be read from a file and/or from direct input in a way analogous to the nodal temperature input.

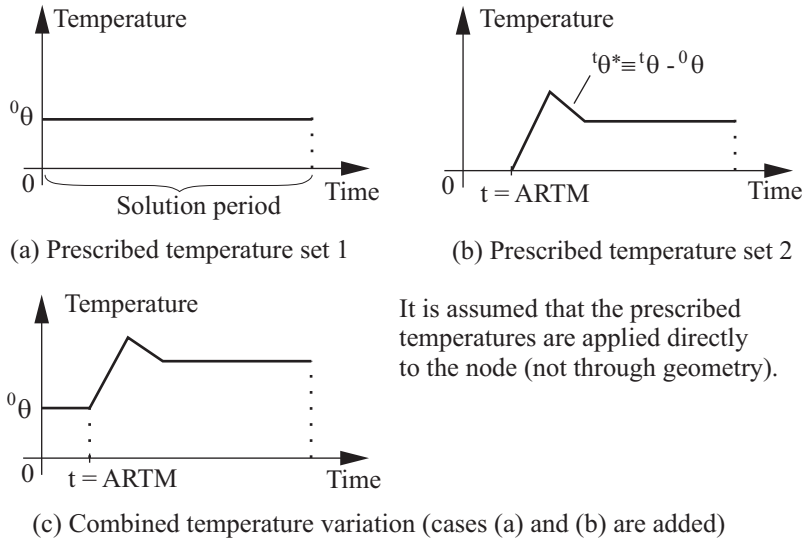
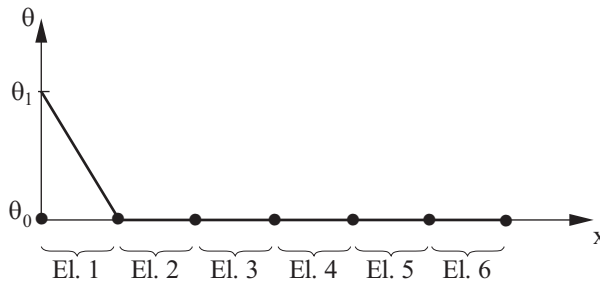


Figure 5.6-1: Use of the arrival time option for prescribed temperatures

- The input of prescribed temperatures and temperature gradients is completely separate from the input of initial temperatures and temperature gradients. See Section 11.6 for the input of initial temperatures and temperature gradients.
- It should also be noted that when using higher-order elements, the temperatures can be significantly different within the element than at the nodal points. For example, the temperature can be negative at points within an element, although the nodal point temperatures are all non-negative. This is illustrated in Fig. 5.6-2 (see element 1). This observation can be important when performing an analysis with temperature-dependent material properties.
- Prescribed temperatures and temperature gradients are not allowed in explicit analysis. In explicit analysis, the initial temperature distribution/reference is used for all solution times in the analysis. When restarting from static or implicit dynamic analysis to explicit analysis, the temperature distribution/reference at the restart time is used.

Linear elements:



Quadratic elements:

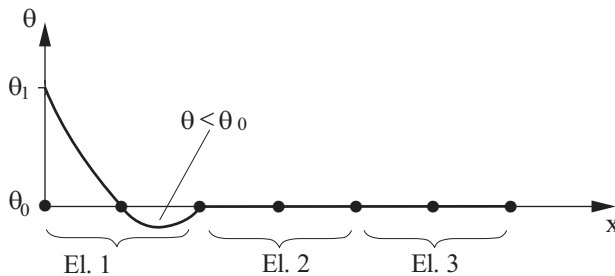


Figure 5.6-2: Interpolation of temperature boundary conditions

5.7 Pipe internal pressure data

- If pipe elements are used, then pipe internal pressures can be input as a load on the structure (see Section 2.8.2). The pipe internal pressure is a nodal point variable which must be defined for all nodes throughout the solution period.
- The time variation of the nodal values of the internal pressure is specified by the use of time functions.
- Pipe internal pressures can also be read from a separate user-supplied file. The format of this file is analogous to the file format used for reading nodal temperatures (see Section 5.6).
- If pipe internal pressures are specified by time functions and also read from a file, the final pipe internal pressures are obtained by adding the separate contributions.

5.8 Electromagnetic loading

- The interaction effects of short-circuit currents in cables and beams can be modeled with ADINA using the electromagnetic loading option. Cables and beams can be modeled with truss and beam elements.
- Considering two electric conductors traversed by short-circuit currents $i_1(t)$ and $i_2(t)$ and placed in a medium of magnetic permeability μ_0 , each differential element $d\mathbf{r}_2$ of one conductor exerts on a differential element $d\mathbf{r}_1$ of the other conductor a magnetic force given by Grassman's formula (see Fig. 5.8-1):

$$d\mathbf{F}_{21} = \frac{\mu_0}{4\pi} i_1(t) i_2(t) \frac{d\mathbf{r}_1 \times (d\mathbf{r}_2 \times \mathbf{r}_{21})}{\|\mathbf{r}_{21}\|^3}$$

where \mathbf{r}_{21} is the vector between the middle of the differential elements $d\mathbf{r}_1$ and $d\mathbf{r}_2$. Integration of $d\mathbf{F}_{21}$ along a finite element yields the total electromagnetic force on the element, half of which is assigned to each end node.

- When entering this loading in the AUI, it is necessary to apply the load onto each of the electric conductors. For the load on cable k , the time function value should equal $\sqrt{\frac{\mu_0}{4\pi}} i_k(t)$.
- 3-point Gauss integration is used for all numerical integrations.

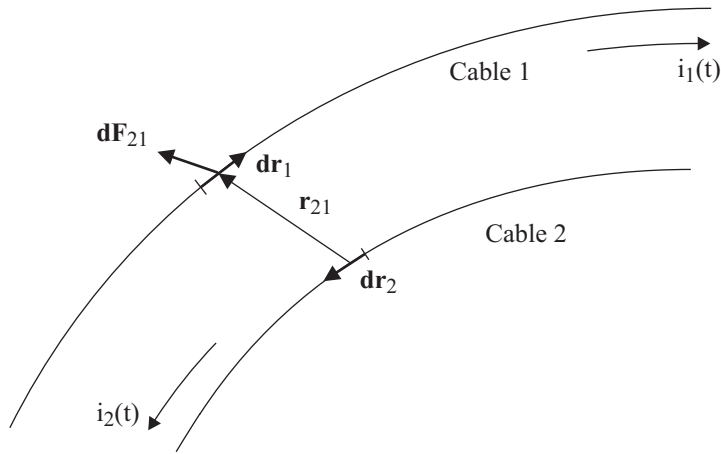


Figure 5.8-1: Conventions for electromagnetic loading

5.9 Poreflow loads

- The flux of pore fluids in porous media can be specified using poreflow loads. See Section 3.12 for more information about analysis of porous media.
- The flux can either be concentrated (that is, applied directly to nodes), or distributed (that is, applied to the boundaries of 2-D and 3-D elements).

5.10 Phiflux loads

- The flux of fluids in potential-based fluid elements can be specified using phiflux loads. See Section 2.11 for more information about the potential-based fluid elements and Section 2.11.14 for more information about loads applied to potential-based fluid elements.
- The flux can either be concentrated (that is, applied directly to nodes), or distributed (that is, applied to the boundaries of 2-D and 3-D elements).
- Positive mass flux is assumed to represent mass flowing into the fluid domain.

5.11 Contact slip loads

- The contact slip load enables you to add a prescribed velocity to the contact surface. Conceptually, the contact surface is assumed to be undergoing a prescribed rotation with a given axis of rotation, an angular velocity and a time function. The velocity corresponding to the prescribed rotation is termed the contact slip loading velocity.

The total velocity of a contact surface is the vectorial sum of the velocity from the nodes and the contact slip loading velocity $\boldsymbol{\omega} \times \mathbf{r}$. This vectorial sum is shown in Figure 5.11-1. If the nodes are fixed, then the total velocity of the contact surface is obtained entirely from the contact slip loading velocity.

The velocity used in friction calculations is the total velocity of the contactor node (including contributions from any slip velocity loading applied to the contactor surface) minus the total velocity of the target segment (including contributions from any slip velocity loading applied to the target surface).

- Contact slip loading can be applied to 2-D and 3-D contact surfaces.
- In 2-D analysis, the slip load velocity vector can be out of the plane of the FE model, for example, in the x direction if the model is defined in the y-z plane.

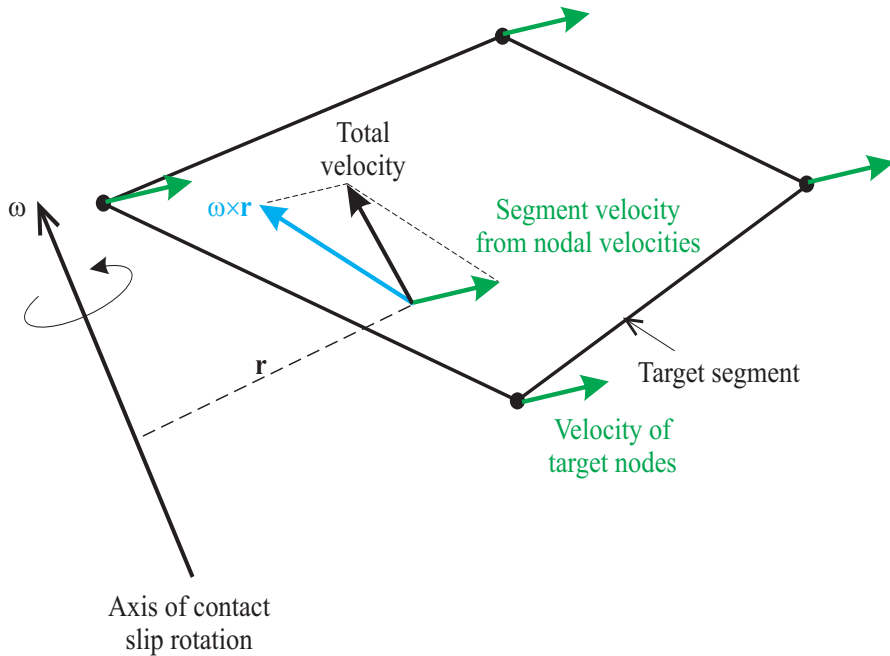


Figure 5.11-1: Contact slip loading

5.12 Surface tension boundary

- A constant surface tension value can be applied to the boundaries of 2-D and 3-D elements as shown in Figure 5.12-1. A similar boundary condition for fluids exists in ADINA-F.

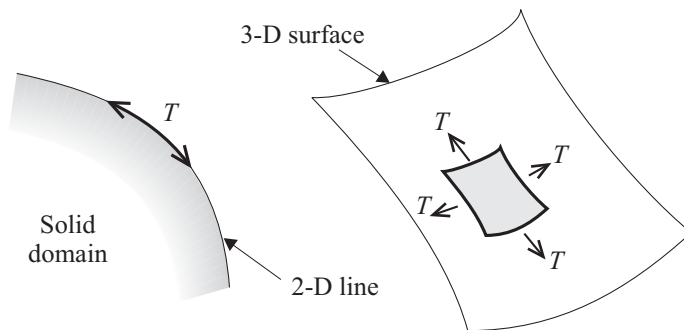


Figure 5.12-1: Surface tension applied to 2-D and 3-D boundaries

- In 2-D analysis, the surface tension boundary can be applied to geometry lines or element edges. In 3-D analysis, the surface tension boundary can be applied to geometry surfaces or element faces.
- The surface tension boundary is converted by the AUI into equivalent ADINA elements when the ADINA data file is generated. The ADINA elements have an initial stress equal to the magnitude of the surface tension, T . In 2-D analysis, isobeam elements are used. In 3-D analysis, 2-D generalized plane stress elements are used. When post-processing the results, the AUI automatically converts the ADINA elements back into a surface tension boundary.
- The surface tension option can be used with the **small displacement/small strain**, **large displacement/small strain** and **large displacement/large strain** formulations. When large displacements are selected, the surface tension is treated as a true (Cauchy) stress.

5.13 Initial load calculations

- Considering temperature, temperature gradient and pipe internal pressure loads, see Section 11.6 where the initial conditions are discussed.
- Considering all other externally applied loading, the loads corresponding to time TSTART are not used and not calculated, except when the ATS method is employed. When the ATS method is employed, the loads are calculated either directly from the time functions, or, in the case of fluid-structure interaction with ADINA-FSI, from the initial displacements, initial velocities, initial accelerations and initial strains.

5.14 User-supplied loads

- An option for user-supplied loads is available in ADINA to allow the modeling of very general types of loading. For example, loads can be defined as functions of nodal coordinates,

displacements, velocities, accelerations, temperatures, times, etc.

5.14.1 General considerations

- In order to specify the loads, you must provide the desired definitions of the loads in the form of FORTRAN coding which is inserted in the specially provided subroutines in ADINA.
- The user-supplied loads applied at a node can be a function of the various nodal quantities at the same node, or they can be a function of nodal quantities at other nodes, as described in detail below.
- The presence of user-supplied loads requires in general the use of equilibrium iterations in the response solution, see Section 7.2.

5.14.2 Usage of the user-supplied loads

- User-supplied loads can be specified in static or dynamic, linear or nonlinear analysis. However, the definition of the loads must be consistent with the type of analysis; e.g., the loads cannot be a function of the velocities in a static analysis.
- User-supplied loads cannot be specified if the LDC method is employed (see Section 7.2.1 for information about the LDC method). In addition, user-supplied loads cannot be applied to any substructures; however, these loads can be applied to the main structure to which substructures are coupled.
- During the step-by-step response calculation, the user-supplied loads are evaluated in each time step and in each iteration. These loads are added to the effective load vector normally established in the response calculation for the time steps.

Evaluation of the user-supplied loads: You must modify the coding in the user-supplied subroutines IUSER and USERSL in order to select the nodes whose nodal variables are used in the load calculation, to select the nodes to which the loads are applied, and to calculate the loads.

Sample coding is provided in file ovl170u.f. This coding corresponds to the example given in Section 5.14.3 below.

- You enter some control parameters in the AUI in the specification of the user-supplied loading (in the command-line interface, the command is `APPLY USER-SPECIFIED-LOADS`). If loads calculated at a given node *M* depend only upon variables at the node *M*, set parameter `NODE-DEPENDENCE` to 1; if loads calculated at a given node *M* depend upon variables at other nodes, set parameter `NODE-DEPENDENCE` to 2.

You also can define integer and floating point constants in the AUI for use in the user-supplied loading subroutines:

NRCONS: Number of floating point constants (maximum of 100).

NCONS: Number of integer constants (maximum of 100).

RCONS(NRCONS): Floating point constants. In subroutine `USERSL`, this corresponds to vector `ULDATA(100)`.

ICONS(NCONS): Integer constants. In subroutine `USERSL`, this corresponds to vector `KULINT(100)`.

NODE-DEPENDENCE=1: If `NODE-DEPENDENCE=1`, the loads applied at a node *M* can depend only on nodal variables at the same node.

Node selection using subroutine `IUSER`: Subroutine `IUSER` is called once for each nodal point each time the effective load vector is calculated. In this case `MFLAG` should be set to 1 only for nodes at which loads are to be applied.

Load calculation using subroutine `USERSL`: Subroutine `USERSL` is called once for each node selected by subroutine `IUSER` each time the effective load vector is calculated.

Note that for each node *M*, only the non-zero components of the new additional load vector need to be calculated. ADINA distinguishes internally independent, dependent or fixed degrees of freedom and performs the proper summation of the new additional load vector to the effective load vector.

NODE-DEPENDENCE=2: If NODE-DEPENDENCE=2, then the loads applied at a node M depend on the nodal variables at a number of other nodes N.

Node selection using subroutine IUSER: Two flags, MFLAG and NFLAG, are used in subroutine IUSER to select the appropriate nodes for the load calculation, as schematically shown.

```
DO 100 M=1, NUMNP
DO 200 N=1, NUMNP
.
.
.
CALL IUSER(M, N, NUMNP, MFLAG, NFLAG)
IF (MFLAG.EQ.0) GO TO 100
IF (NFLAG.EQ.0) GO TO 200
.
.
.
CALL USERSL(...)
.
.
.
200 CONTINUE
100 CONTINUE
```

Note that subroutine USERSL is only called for nodes selected by the flags set in subroutine IUSER.

Load calculation using subroutine USERSL: Each time the effective load vector is calculated, for each node M at which the user-supplied loads are applied, subroutine USERSL is called once for $N=N_1, N_2, \dots, N_N$ where N_1, N_2, \dots, N_N are arranged in ascending order and correspond to the nodes whose nodal variables are used to calculate the loads at node M. Nodes N_1, N_2, \dots, N_N are specified in subroutine IUSER as described above.

Note that for the load calculations at node M, the required nodal variables must be stored in auxiliary working arrays when $N.LT.NN$, and the actual load computation must be performed only when $N.EQ.NN$. The arrays WA(6), WB(6), ... WF(6) are

provided for this purpose.

Note that for each node M, only the nonzero components of the additional load vector need to be calculated. ADINA distinguishes internally independent, dependent or fixed degrees of freedom and performs proper summation of the new additional load vector to the effective load vector.

5.14.3 Example: Hydrodynamic forces

- For a cylinder moving with velocity $\dot{\mathbf{U}}_c(t)$ and acceleration $\ddot{\mathbf{U}}_c(t)$ in the direction of an oncoming fluid stream of velocity $\dot{\mathbf{U}}_f(\mathbf{x}, t)$, the total hydrodynamic force per unit length due to drag and inertial effects, excluding diffraction effects, can be approximated by Morison's general formula:

$$\begin{aligned} \mathbf{F}_c = & 0.5D\rho C_D (\dot{\mathbf{U}}_f - \dot{\mathbf{U}}_c) \|\dot{\mathbf{U}}_f - \dot{\mathbf{U}}_c\| \\ & + 0.25\pi D^2 \rho (1 + C_A) \left[\frac{\partial \dot{\mathbf{U}}_f}{\partial t} + (\dot{\mathbf{U}}_f - \dot{\mathbf{U}}_c) \cdot \frac{\partial \dot{\mathbf{U}}_f}{\partial \mathbf{x}} \right] \\ & - 0.25\pi D^2 \rho C_A \ddot{\mathbf{U}}_c \end{aligned} \quad (5.14-1)$$

where ρ is the density of the fluid, D is the diameter of the cylinder and C_D and C_A are the drag and added mass coefficients of the cylinder.

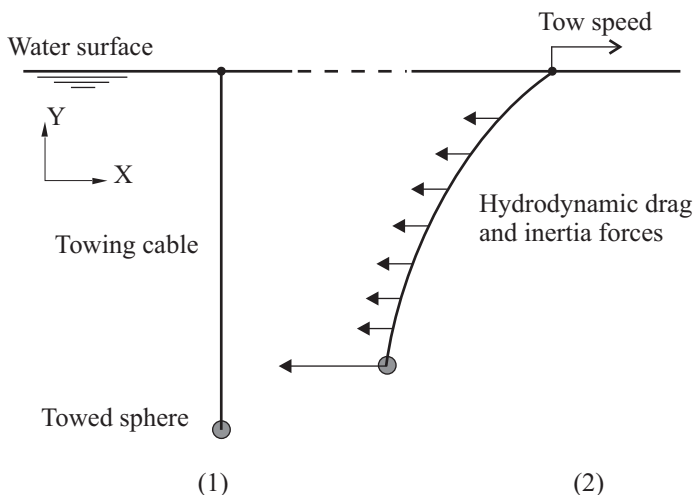
In the right-hand-side of (5.14-1), the first term represents the form drag, the second term corresponds to inertial forces due to the local and convective accelerations of the fluid flowing around the cylinder and the third term corresponds to the inertial force due to the motion of the cylinder.

Morison's equation is commonly used for the analysis of submerged and partially submerged structures such as offshore rigs, risers, underwater pipes and cables.

More information about Morison's equation and about the drag and inertia coefficients C_D and C_A (which depend upon many parameters, including the section shape and surface roughness of the cylinder and the Reynolds number) can be found in the following reference:

ref. Sarpkaya, T. and Isaacson, M., *Mechanics of Wave Forces on Offshore Structures*, Van Nostrand Reinhold Co., New York, 1981.

- In ADINA, the user-supplied load interface has been used to implement the calculation of the Morison forces in the case of a sphere suspended at the bottom of a cable, and towed in a still fluid $\dot{\mathbf{U}}_f(\mathbf{x}, t) = \mathbf{0}$ (see Figure 5.14-1). The model must consist of one line of elements, with possibly concentrated masses and/or dampers at the nodes. The nodes must be numbered 1, 2, \dots , N . At the bottom node N , a sphere generating additional drag and inertia forces can be added. The following items can be specified: the length of the cable, the fluid density, the diameter of the sphere, the cable drag and added mass coefficients and the sphere drag and added mass coefficients.



- (1) Initial geometry
 (2) Structure submitted to gravity and hydrodynamic forces

Figure 5.14-1: Usage of the user-supplied load subroutines for the analysis of a sphere towed underwater

- The hydrodynamic force on a unit length of the cylinder (velocity $\dot{\mathbf{U}}_c$, acceleration $\ddot{\mathbf{U}}_c$) is calculated using

$$\mathbf{F}_c = -0.5D\rho C_D \dot{\mathbf{U}}_c \|\dot{\mathbf{U}}_c\| - 0.25\pi D^2 \rho C_A \ddot{\mathbf{U}}_c$$

where ρ is the density of the fluid, D is the diameter of the cylinder and C_D and C_A are the drag and added mass coefficients of the cylinder.

- The hydrodynamic force on the sphere (velocity $\dot{\mathbf{U}}_s$, acceleration $\ddot{\mathbf{U}}_s$) is calculated using

$$\mathbf{F}_s = -0.5\left(\frac{D}{2}\right)^2 \rho C_D \dot{\mathbf{U}}_s \|\dot{\mathbf{U}}_s\| - \frac{4}{3}\pi\left(\frac{D}{2}\right)^3 \rho C_A \ddot{\mathbf{U}}_s$$

where ρ is the density of the fluid, D is the diameter of the sphere and C_D and C_A are the drag and added mass coefficients of the sphere.

- Either a static or a dynamic analysis can be performed with ADINA. In a static analysis, the velocity at the top of the cable must be entered.
- Verification problem B.130 in the ADINA Verification Manual demonstrates the usage of this user-supplied load.

5.15 Constraint equations

5.15.1 General considerations

- ADINA can impose two basic types of constraints between nodal degrees of freedom. The first type has the form

$$U_k = \sum_j \beta_j U_j \quad (5.15-1)$$

where U_k is a dependent (or slave) degree of freedom which is

controlled by multiple independent (master) U_j degrees of freedom via factors β_j . Alternatively, a general constraint can be defined as

$$\sum_j \beta_j U_j = 0 \quad (5.15-2)$$

where none of the U_j degrees of freedom are made dependent (there are no slave degrees of freedom). In this case, the constraint is imposed using Lagrange Multipliers.

Note that each constraint of the first type reduces the number of independent degrees of freedom by one, while each general constraint of the second type increases the number of degrees of freedom by one (by adding a Lagrange Multiplier). Hence, the first type should be used whenever possible. In some cases, however, one cannot easily express a constraint in the form of Eq. (5.15-1) where a slave degree of freedom is only related to master degrees of freedom. See the example below for an illustration.

$$\begin{aligned} U_{2(z)} &= U_{1(z)} + 3\theta_{1(x)} && \leftarrow \text{valid constraint} \\ U_{3(z)} &= U_{2(z)} + U_{1(z)} - 4\theta_{2(x)} && \leftarrow \text{invalid constraint} \\ &\uparrow \\ &\text{This DOF is already dependent} \end{aligned}$$

- Note that in the above example, the problem with the second constraint could be resolved by manually applying the first constraint to obtain

$$U_{3(z)} = 2U_{1(z)} + 3\theta_{1(x)} - 4\theta_{2(x)}$$

- The mesh glueing feature (Section 5.16) automatically produces general constraints which are expressed in the form of Eq. (5.15-2).
- Displacement constraints can be applied to static and dynamic analyses.
- The basic constraints of Eq. (5.15-1) are only approximately satisfied in an explicit analysis, since imposing the constraint

exactly requires a non-diagonal mass matrix.

- General constraints cannot be used in explicit analysis.
- The following comments apply to basic constraints of Eq. (5.15-1) in which the slave degree of freedom is constrained to more than one master degree of freedom.
 - If a consistent mass matrix is used and concentrated masses are applied to constrained nodes, then the off-diagonal terms are included in the global mass matrix.
 - Similarly, if concentrated dampers are applied to constrained nodes, then the off-diagonal terms are included in the global damping matrix.

5.15.2 Rigid links

- Rigid links are special constraint equations established automatically by the program between two nodes — a master node and a slave node.

As the nodes displace due to deformation, the slave node is constrained to translate and rotate such that the distance between the master node and the slave node remains constant, and that the rotations at the slave node are the same as the corresponding rotations at the master node.

- Rigid links can be kinematically linear or nonlinear. For linear rigid links, the constraint equations are unchanged throughout the solution and hence they are applicable for small displacement analysis.

In a large displacement analysis, nonlinear rigid links should be used recognizing that the effects of large displacements are taken into consideration in updating the constraint equations for the master and slave nodes. Hence, the coefficients in Eq. (5.15-1) are no longer constant.

- It is allowed to select the degrees of freedom that are selected by a rigid link. By default, all master degrees of freedom are constrained.

- Rigid links are useful in the modeling of stiffened shells, for example, in coupling iso-beam elements (see Section 2.5) to shell elements (see Section 2.7).
- The internal implementation of the tied contact feature discussed in Section 4.14 is conceptually similar to creating rigid links between the contact surfaces.
- When a rigid link that constrains rotational degrees of freedom is attached to a node of a shell element, 6 degrees of freedom must be used for the node.

The rotation around the shell node director vector may have zero stiffness, depending upon the modeling of the other end of the rigid link (see Figure 5.15-1). There is an option to assign stiffness to those shell node rotational degrees of freedom with zero stiffness that are attached to rigid links.

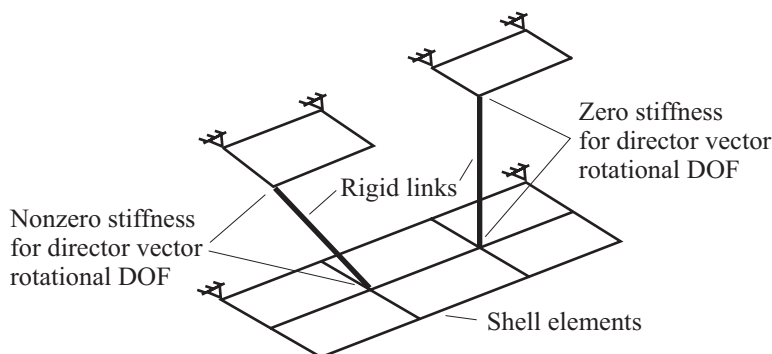


Figure 5.15-1: Rigid links and shell elements

- You can use fixities on nodes connected to rigid links, provided that the fixities not prevent the rigid link from maintaining constant distance between the nodes.
- When a large displacement rigid link is attached to a linear element or a deformation-independent load, the initial constraint equations are used.

Rigid links and continuum elements: Care must be used when connecting continuum elements (2-D and 3-D solid elements) with rigid links. In order for the rigid link to rotate, the master DOF of the rigid link must have its rotational DOFs free.

This situation is illustrated in Figure 5.15-2. Two groups of 2-D solid elements are connected together with large displacement rigid links. If the x rotational DOFs are fixed, then when the lower element is rotated, the rigid links maintain both their length and their angle, and the upper element group distorts as shown in Fig 5.15-2(b). If the x rotational DOFs are free, then when the lower element is rotated, the rigid links are free to rotate and the upper element group can also rotate without distortion as shown in Fig 5.15-2(c).

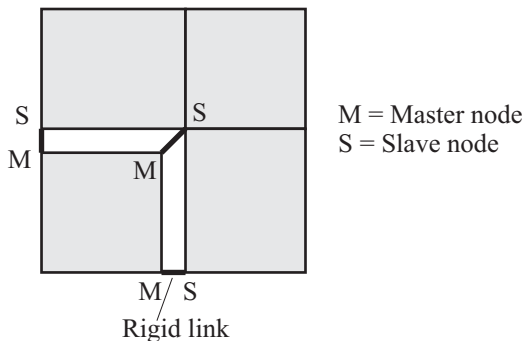
By default, the AUI does not fix the rotational DOFs of all nodes connected to continuum elements. Thus, by default, the model in Figure 5.15-2 will behave as shown in Fig 5.15-2(c), as expected. However, this default can be changed using the command parameter

ADINA ... FIXBOUNDARY=V89

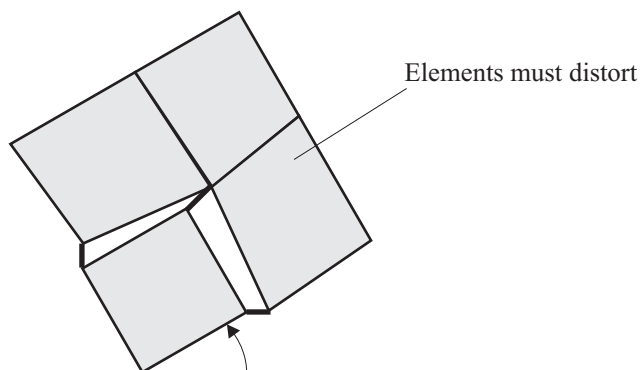
or

KINEMATICS ... RIGIDLINK-6DOF=V89

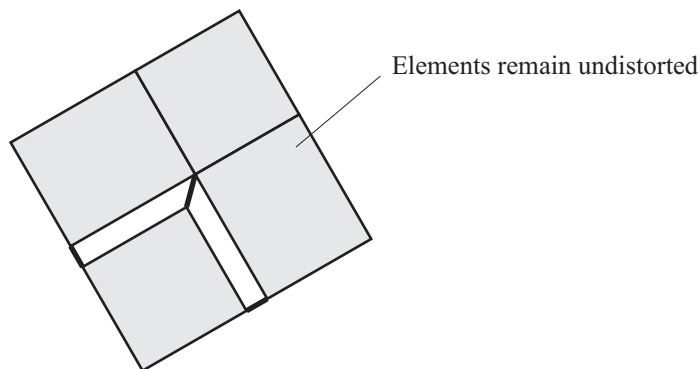
With this option, the AUI fixes the rotational DOFs of nodes connected to continuum elements and rigid links, and the model in Figure 5.15-2 will behave as shown in Fig 5.15-2(b). This option is recommended for use only for backwards compatibility.



a) Two element groups connected with rigid links



b) Master rotational DOFs are fixed, rigid links cannot rotate



c) Master rotational DOFs are free, rigid links can rotate

Figure 5.15-2: Rigid links used with continuum elements

5.15.3 General rigid links

As its name implies, the general rigid link makes use of general constraints rather than basic constraints.

General rigid links can be defined in the same manner as basic rigid links. However, compared to the basic constraint rigid link, the general constraint rigid link has the following capabilities:

- ▶ General constraint rigid links may be chained
- ▶ Fixities and prescribed displacements may be applied to slaves and master nodes
- ▶ Basic constraints and general constraints can be applied to general constrain rigid links
- ▶ Mesh glue can be applied on slave and master nodes
- ▶ The general rigid link uses a compliance, or inverse stiffness, to enforce its internal constraints

General rigid links are always geometrically nonlinear. Different skew systems may be defined on the master and slave nodes.

5.15.3.1 Theory

Consider a rigid link with master and slave nodes. Given the current position of the master node \mathbf{Y} , the kinematic relations give an expected slave node position \mathbf{Z} . The slave's actual current position is \mathbf{X} . Fig. 2.15.4-1 shows that the difference between the actual and expected slave node positions is the gap $\mathbf{X} - \mathbf{Z} = \mathbf{g}$. Also, let \mathbf{v} be the displacements of the master node and \mathbf{u} be the displacements of the slave node.

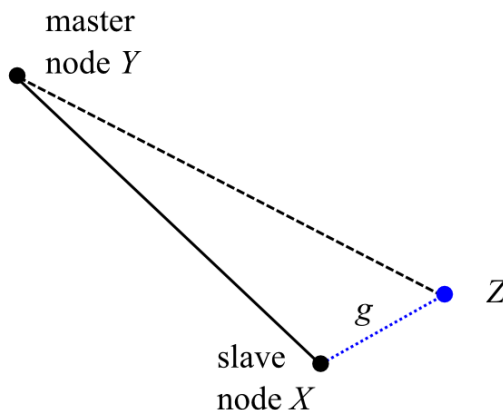


Figure 2.15.4-1: General rigid with master node at position Y and slave node at position X . Given the motions of the master node, the expected position of the slave node is Z . The distance between the expected and actual position of the slave node is the gap g .

The constraint to be enforced is $X - Z = \mathbf{0}$, or $\mathbf{g} = \mathbf{0}$. The forces due to motions at the gap are

$$\bar{F}_i = -\frac{1}{\hat{c}} g_i,$$

where $i = 1, 2, 3$ is for the translations, and $i = 4, 5, 6$ is for the rotations, and \hat{c} are the rigid link compliances for the constrained translational or rotational degrees of freedom.

The general rigid link constraint is based on the difference between the forces due to displacements (and moments due to rotations) at the gap and the separately computed forces

$$\delta F_i (F_i - \bar{F}_i) = 0 \Rightarrow \delta F_i \left(F_i + \frac{1}{\hat{c}} g_i \right) = 0,$$

where \hat{c} are the compliances corresponding to each gap. The constraint is multiplied by \hat{c} to obtain

$$\delta F_i (\hat{c} F_i + g_i) = 0,$$

which recovers the expected constraint as the compliances approach zero.

Assuming small compliances, we can replace $\hat{c} F_i + g_i$ with g_i . The compliances do not appear in right-hand side of the system. Rather, the constraints are only satisfied when their associated gaps are zero. As a result, the general rigid link is truly rigid, and for a tightly converged solution, the gaps are always zero.

However, because the compliances appear in the left-hand side of the system but not in the right-hand side, the solution will not be quadratic. Thus the choice of compliance affects only the convergence rate, and not the converged solution. This is similar to how stiffness stabilization works for nonlinear problems.

5.15.3.2 Modeling

General rigid links can be generated manually by using the `RIGIDLINK TYPE=GENERAL` command. By default, the `RIGIDLINK TYPE` option is set by `RL-CONTROL TYPE=AUTOMATIC`. Generalized constraints are created for rigid links that are chained or have fixities, and basic constraints are created for all other rigid links. In other words, by default, general rigid links can be generated automatically when necessary, with a preference for generating basic constraint rigid links whenever possible.

As with the basic rigid link, the user must specify the slave degrees of freedom (DOF) to be constrained to the master node. `DOFSI` must contain 1 to 6 digits ranging from 1 to 6. DOFs 1, 2, 3 indicate X, Y, Z translations and 4, 5, 6 indicate X, Y, Z rotations.

5.15.3.3 Notes and Recommendations

- 1) The general rigid link cannot be used in explicit dynamic analysis or cyclic symmetry analysis.
- 2) In problems involving general rigid links undergoing large rotations, it is recommended to use the nonsymmetric sparse solver (MASTER SOLVER = NONSYM-SPARSE).
- 3) The default value for the translational and rotational compliances RLTVALUE and RLRVALUE of the general rigid link are 1.0×10^{-6} , which may be too large in some instances. If a general rigid link is not enforcing the constraints sufficiently well, consider using a smaller value for the compliance. Note that as the compliance factor becomes smaller, the convergence rate will decrease, as expected.
- 4) The general rigid link does not connect temperature degrees of freedom.
- 5) The general rigid link cannot be used on target contact surfaces when the rigid target algorithm is used. This restriction does not apply when using the default constraint function contact algorithm.
- 6) The general rigid link has no mass.
- 7) In frequency analysis, the compliances will affect the results.

5.15.4 RBE3 elements

- The RBE3 element defines the motion of a reference node as a weighted average of the motion of a set of other nodes. This element is a useful tool for distributing applied load and mass in a model.
- The reference node is denoted as slave node, and only the specified DOFs (SLAVEDOF in AUI command RBE3) of the slave node are connected to the RBE3 element. Other nodes in the RBE3 element are denoted as master nodes, and only the specified DOFs (dof_i in AUI command RBE3) of the master nodes are connected to

the RBE3 element.

- The element is implemented as a set of constraint equations in which the motions of the slave node are constrained to the motions of the master nodes.
- In the following, we briefly outline the derivation of the RBE3 element. For simplicity, we consider just the case in which all components are given in the basic coordinate system, but the RBE3 element allows for the components of the slave node, and for the components of each of the master nodes, to all be in a different coordinate system.

Let superscript q denote the slave node (SLAVENAME in AUI command RBE3) and superscript k denote one of the master nodes (mastername _{i} in AUI command RBE3), with associated weight (w_i in AUI command RBE3). The derivation is based on the transmission of forces/moments from the master nodes k to the slave node q , and on the transmission of forces/moments from the slave node q to the master nodes k .

Firstly, if the forces/moments at a given master node k are known, then these forces are transmitted to slave node q using the equilibrium equation.

$$\begin{bmatrix} F_x^q \\ F_y^q \\ F_z^q \\ M_x^q \\ M_y^q \\ M_z^q \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -(z^k - z^q) & (y^k - y^q) & 1 & 0 & 0 \\ (z^k - z^q) & 0 & -(x^k - x^q) & 0 & 1 & 0 \\ -(y^k - y^q) & (x^k - x^q) & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} F_x^k \\ F_y^k \\ F_z^k \\ M_x^k \\ M_y^k \\ M_z^k \end{bmatrix}$$

or, in matrix form,

$$\mathbf{F}^q = (\mathbf{S}^k)^T \mathbf{F}^k \quad (5.15-3)$$

where \mathbf{F}^q and \mathbf{F}^k are column vectors containing the forces and moments, and \mathbf{S}^k is the transpose of the square matrix in the above equation.

Therefore if the forces/moments at all master nodes k are known, then the forces/moments at slave node q is found using

$$\mathbf{F}^q = \sum_k (\mathbf{S}^k)^T \mathbf{F}^k \quad (5.15-4)$$

Secondly, we now postulate that, if the forces/moments at slave node q are known, that these forces/moments are transmitted to master node k using

$$\mathbf{F}^k = \mathbf{W}^k \mathbf{S}^k \mathbf{X} \mathbf{F}^q \quad (5.15-5)$$

where $\mathbf{W}^k = \begin{bmatrix} w_1^k & & & & & \\ & w_2^k & & & & \\ & & w_3^k & & & \\ & & & w_4^k & & \\ & & & & w_5^k & \\ & & & & & w_6^k \end{bmatrix}$ is a diagonal

matrix with weighting factors given by wt_i and dof_i as follows:

For each component l , $l=1,2,3,4,5,6$, then

 If component l is not in the dof_i list, then

$$w_l^k = 0$$

 else

 if $l=1,2,3$, then

$$w_l^k = wt_i$$

 else

$$w_l^k = wt_i \cdot L_c^2$$

 endif

end

In this calculation, L_c is the average distance between the slave node and all of the master nodes. If this average distance is zero,

then $L_c = 1$. L_c is introduced into the weighting matrix so that the element has dimensional independence.

Matrix \mathbf{X} is a 6x6 matrix that is the same for all of the connected nodes. Matrix \mathbf{X} is determined as follows. Combining (5.15-4) and (5.15-5) gives

$$\mathbf{F}^q = \sum_k (\mathbf{S}^k)^T \mathbf{W}^k \mathbf{S}^k \mathbf{X} \mathbf{F}^q \quad (5.15-6)$$

and for this to be satisfied for all \mathbf{F}^q ,

$$\mathbf{X} = \left(\sum_k (\mathbf{S}^k)^T \mathbf{W}^k \mathbf{S}^k \right)^{-1} \quad (5.15-7)$$

Now let

$$\mathbf{G}^k = \mathbf{W}^k \mathbf{S}^k \mathbf{X} \quad (5.15-8)$$

From (5.15-5),

$$\mathbf{F}^k = \mathbf{G}^k \mathbf{F}^q \quad (5.15-9)$$

Finally, we apply the principle of virtual work to the RBE3 element:

$$\begin{aligned} (\mathbf{F}^q)^T \delta \mathbf{u}^q &= \sum_k (\mathbf{F}^k)^T \delta \mathbf{u}^k \\ &= \sum_k (\mathbf{F}^q)^T (\mathbf{G}^k)^T \delta \mathbf{u}^k \end{aligned} \quad (5.15-10)$$

(in which $\delta \mathbf{u}$ are the virtual displacements/rotations), and for (5.15-10) to be satisfied for all \mathbf{F}^q ,

$$\delta \mathbf{u}^q = \sum_k (\mathbf{G}^k)^T \delta \mathbf{u}^k \quad (5.15-11)$$

(5.15-11) shows that the virtual displacements/rotations of the slave node must be related to the virtual displacements/rotations of the master nodes. This can only occur if the actual

displacements/rotations of the slave node are related to the actual displacements/rotations of the master nodes:

$$\mathbf{u}^q = \sum_k (\mathbf{G}^k)^T \mathbf{u}^k \quad (5.15-12)$$

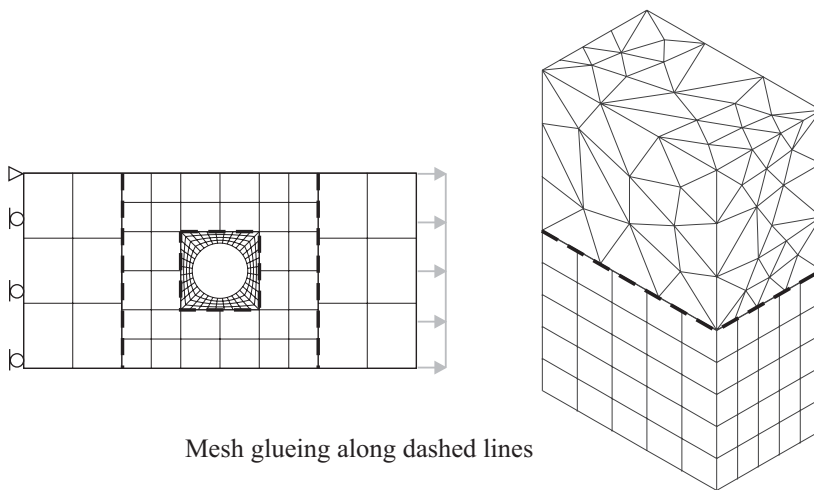
(5.15-12) has six rows. Each row l selected by a component in REFC corresponds to a multipoint constraint equation, in which u_l^q is dependent and u^k is independent. Rows not selected by a component in REFC are discarded.

- The RBE3 element assumes that the displacements of the slave and master nodes are small. This is seen in the definition of matrix \mathbf{S}^k , since the original coordinates of the nodes are used in this matrix.
- The RBE3 element applies a basic constraint to the slave node such that the slave degrees of freedom are dependent on the master degrees of freedom. The RBE3 element specifies the slave degrees of freedom to be constrained and the master degrees of freedom used for the constraint. The weighting factors of the master degrees of freedom can only be adjusted if the master entities are defined using nodes or points.
- The RBE3 element only constrains the displacement degrees of freedom (translations and rotations). All other degrees of freedom such as pipe ovalization, warping, and fluid potential, etc. are not constrained by the RBE3 element.
- In general, it is recommended that only the RBE3 master translational displacement degrees of freedom are used (MASTERDOF=123). The RBE3 master rotational displacement degrees of freedom should be used in special applications, such as when all master nodes are collinear.
- If the RBE3 element is used to distribute load or mass from the slave node to the master nodes, the RBE3 element only applies load or mass to the master degrees of freedom. For example, if MASTERDOF=2, then the RBE3 element only applies Y-direction load or mass to the master nodes.

- Different skew systems may be assigned to the master and slave nodes.
- If a RBE3 master node is a shell node, the AUI will only switch it to a 6-DOF node if the RBE3 element has rotational DOF for the master nodes.

5.16 Mesh glueing

- The glueing feature is used to attach two surfaces (or lines in 2-D) together. These two surfaces usually involve different finite element meshes (see Fig. 5.16-1). The glueing procedure should lead to smooth transition of displacements and tractions between the glued surfaces. This feature is useful for several applications:
 - ▶ When a fine mesh is desired in a certain region and coarser meshes are desired in other regions.
 - ▶ When different regions are meshed independently with unstructured free meshes.
 - ▶ When different regions are meshed with different element types (such as a tetrahedral mesh attached to a brick mesh).



Mesh glueing along dashed lines

Figure 5.16-1. Examples requiring mesh glueing

- The proper glueing constraint between the two surfaces can be expressed as

$$\int_{\Gamma} \lambda \cdot (u^1 - u^2) d\Gamma = 0 \quad (5.16-1)$$

where u^1 is the displacement of the first glued surface, u^2 is the displacement of the second surface and λ is the Lagrange Multiplier field imposing the constraint.

One of the glued surfaces is designated as the master and the other as the slave. The Lagrange Multiplier field involves nodal degrees of freedom at the nodes of the slave surface, and the integration is also performed over the slave surface. Hence Eq. (5.16-1) becomes

$$\int_{\Gamma^S} \lambda^S \cdot (u^M - u^S) d\Gamma^S = 0 \quad (5.16-2)$$

The accurate integration of Eq. (5.16-2) is not trivial since the displacements u^M and u^S are generally interpolated over different domains. This integration is performed by the AUI and results in multiple general constraints (see Section 5.15.1).

- Mesh glueing is not available in explicit analysis.
- Mesh glueing is available for the ADINA heat flow solution options.
- Glueing is superior to tied contact and should be used in its place whenever applicable.
- Only line-to-line glueing in 2-D analysis, and surface-to-surface glueing in 3-D analysis is supported. The glued elements can have linear or quadratic displacement interpolation.
- Only 2-D solid elements and isobeam (plane strain, plane stress, and axisymmetric) elements can be used in the glueing of 2-D lines. Glueing of 4-node isobeam elements is not allowed.

- Only 3-D solid elements, membrane (3-D plane stress) elements, and shell elements can be used in the glueing of 3-D surfaces. Glueing of 16-node shell elements is not allowed.
- For isobeam elements and shell elements, only the translational degrees of freedom are glued. The rotational degrees of freedom are not glued.
- It is allowed to glue 3-D-solid elements to shell element surfaces.
- Nodes on glued surfaces (both master or slave) cannot have dependent translation degrees of freedom. Therefore, they cannot be slaves in rigid links or constraint equations involving translations.
- If one glue surface is smaller than the other, the smaller surface should preferably be the slave. However, the glueing will also work if the smaller surface is the master. The two glued surfaces can also be partially overlapping. These cases are shown in Figure 5.16-2.

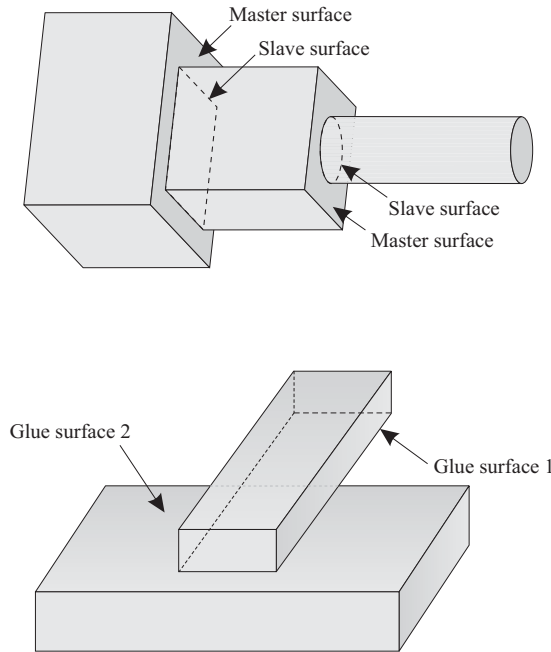


Figure 5.16-2: Examples of master and slave glue surfaces

- The two glued surfaces should ideally be smooth surfaces (no sharp corners). If corners exist it is better to create multiple glued meshes, as shown in Fig. 5.16-3.
- The Lagrange Multiplier field is modified at nodes where multiple glue meshes intersect. These nodes are called cross-over points.

discretization error, as illustrated in Figure 5.16-4.

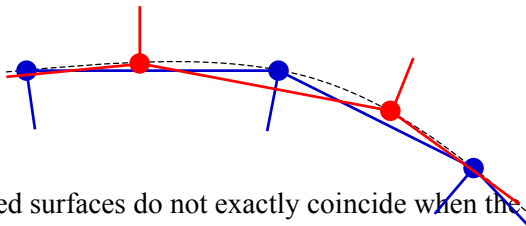
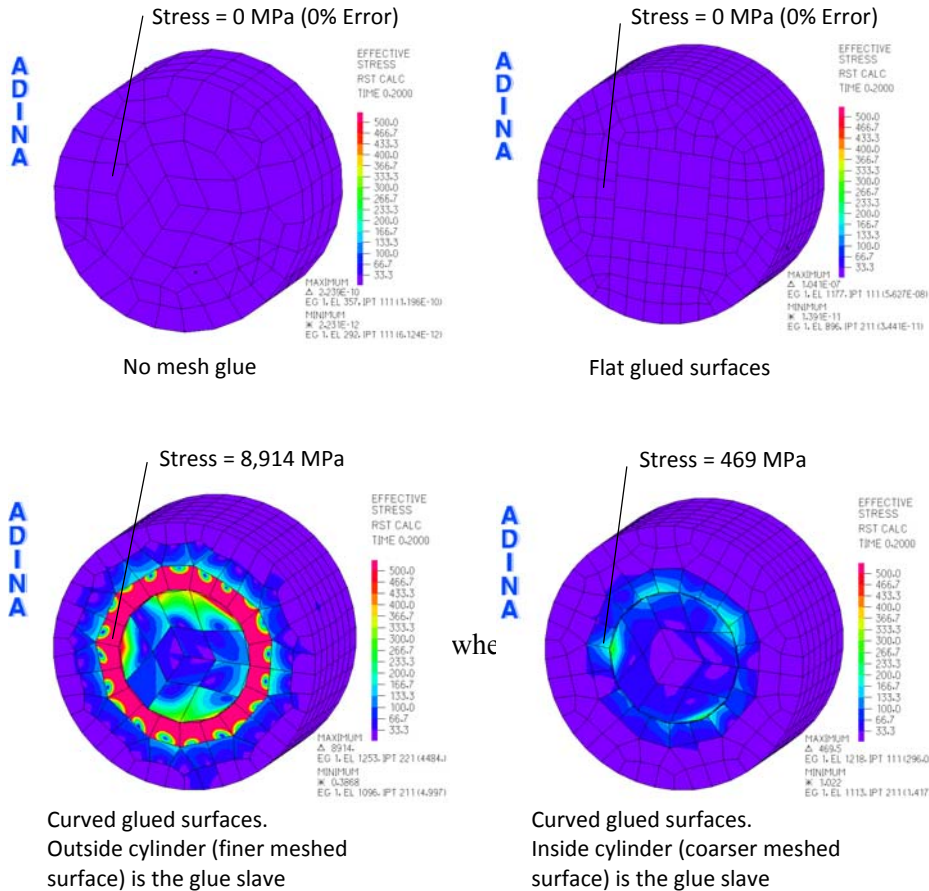


Figure 5.16-4. Glued surfaces do not exactly coincide when the surfaces are curved

As a result, if the glued surfaces are curved, moment equilibrium is not satisfied between the glued surfaces, spurious stresses and strains are generated when the model rotates, and rotational motions are resisted, just as if the glued surfaces are attached to a grounded rotational spring. The stiffer the glued surfaces, the stiffer the rotational spring. This effect is less severe when the coarser meshed surface is the slave surface of the glue mesh pair because in this case the surfaces are not as “tightly” glued (fewer glueing constraints are created), and hence the distortional motion imposed by the mesh glue is less severe.

Figure 5.16-5 shows the stress results for a large displacement static analysis where a steel cylinder of radius 1mm is rotated by 10 degrees by prescribing the rotation on the outside surface of the cylinder. Other than the prescribed rotation, the cylinder is completely free. We observe that the exact solution is obtained when there is no mesh glue and when the glued surfaces are flat, but spurious stresses are generated when the glued surfaces are curved, especially when the finer meshed surface is the slave surface in the mesh glue pair.



5.17 Hydrostatic fluid loading

- ADINA supports hydrostatic fluid loading (or hydro-pressure) for compressible (pneumatic) fluids in situations where the pressure and temperature of the fluid in the cavity is assumed to be uniform at any point in time.
- The hydrostatic fluid model:
 - Can be used to model fluid-filled cavities of compressible (pneumatic) fluids.

- ▶ Does not take the inertia of the fluid into account.
- ▶ Does not require a material definition.
- ▶ Is applicable only for situations where the pressure and temperature of the fluid in the cavity can be assumed to be uniform at any point in time. For cases where a spatially varied pressure and temperature is required, either potential-based fluid elements or ADINA CFD must be used.
- Compressible (pneumatic) fluids are modeled as an ideal gas in isothermal conditions satisfying the law:

$$p_t V_t = f(t) p_0 V_0 \quad (11.19-1)$$

where

p_t = Pressure applied at time, t , by the hydrostatic fluid loading.

V_t = Enclosed volume at time, t .

p_0 = Initial pressure (user-input).

V_0 = Enclosed volume at the birth time, t_B .

$f(t)$ = Time function that can be used to capture the effect of temperature.

- The initial pressure is a user-input.
- For the three-dimensional model, the domain of the fluid-filled cavity can be enclosed by 3D solid element faces, shell elements and 2D elements of membrane case (3D plane stress). The following node numbers of elements or faces are currently supported:
 - 3-node face of 4-node tetrahedral 3D solid elements
 - 3-node triangular 2D solid elements (3D plane stress)
 - 3-node triangular shell elements
 - 4-node face of 8-node hexahedral 3D solid elements
 - 4-node quadrilateral 2D solid elements (3D plane stress)
 - 4-node quadrilateral shell elements

- 6-node face of 10/11-node tetrahedral 3D solid elements
 - 6-node triangular 2D solid elements (3D plane stress)
 - 6-node triangular shell elements
 - 8-node face of 20-node hexahedral 3D solid elements
 - 8-node quadrilateral 2D solid elements (3D plane stress)
 - 8-node quadrilateral shell elements
 - 9-node face of 27-node hexahedral 3D solid elements
 - 9-node quadrilateral 2D solid elements (3D plane stress)
 - 9-node triangular shell elements
 - 16-node quadrilateral shell elements
- For the two-dimensional model, the domain of the fluid-filled cavity can only be enclosed by the edges of 2D solid elements. For plane strain and plane stress cases, since the element thickness is uniform all over the element, the Eq.11.19-1 can be written as:

$$p_t A_t = f(t) p_0 A_0 \quad (11.19-2)$$

where

A_0 = Enclosed area at the birth time, t_B .

A_t = Enclosed area at time, t .

- For the axisymmetric case, it is assumed that the centroid of the cavity does not change along the radial direction during the solution time; therefore, the volume of the cavity is also linearly proportional to the cavity area, and Eq. 11.19-2 is true for this case.
- For a 2D model, either 2 or 3-node edges of the 2D solid elements are supported to enclose the domain of the fluid-filled cavity.
- The Gauss divergence theorem is used to compute the volume of the enclosed domain of the fluid-filled cavity.

For the 3D case, the theorem takes the form

$$\iiint_D (\nabla \cdot \vec{F}) dD = \oiint_S (\vec{F} \cdot \vec{n}) dS \quad (11.19-3)$$

where

D denotes the enclosed domain of the fluid-filled cavity

S denotes the closed surface of the domain

\vec{n} denotes the unit normal vector along the closed surface

If we take the vector field as

$$\vec{F} = \frac{1}{3}(x, y, z) \quad (11.19-4)$$

then,

$$\nabla \cdot \vec{F} = \frac{\partial \vec{F}}{\partial x} + \frac{\partial \vec{F}}{\partial y} + \frac{\partial \vec{F}}{\partial z} = 1$$

and the integration becomes the calculation of the volume of the enclosed domain

$$V = \oint_S (\vec{F} \cdot \vec{n}) dS \quad (11.19-5)$$

For the 2D case, the divergence theorem can be written as

$$\iint_D (\nabla \cdot \vec{F}) dD = \oint_L (\vec{F} \cdot \vec{n}) dL \quad (11.19-6)$$

where

D denotes the 2D enclosed domain of the fluid-filled cavity

L denotes the closed loop of the 2D enclosed domain

\vec{n} denotes the unit normal vector along the closed loop

Assume the model is created in the xy-plane, and we take the vector field as

$$\vec{F} = \frac{1}{2}(x, y) \quad (11.19-7)$$

Then

$$\nabla \cdot \vec{F} = \frac{\partial \vec{F}}{\partial x} + \frac{\partial \vec{F}}{\partial y} = 1 \quad (11.19-8)$$

and the interaction becomes the calculation of the area of the 2D enclosed domain

$$A = \oint_L (\vec{F} \cdot \vec{n}) dL \quad (11.19-9)$$

- Hydrostatic fluid loading supports the feature of birth and death. The birth time is used to specify the initial volume of fluid-filled cavity. The death time is used to remove the hydrostatic fluid loading.

- The definition of hydrostatic fluid loading is similar to that of pressure loading. The both can be applied on surfaces/faces and lines/edges, but note the following for the application of hydrostatic fluid loading:

- ▶ Hydrostatic fluid loading can only be applied once. It's called a hydrostatic fluid loading set.
- ▶ Hydrostatic fluid loading can only be applied on enclosed domains (3D spaces or 2D surfaces). A loading set can be composed of multiple separately enclosed domains.
- ▶ Hydrostatic fluid loading has is physically meaningful only when all the applied pressures are toward the outside of the enclosed domains. During the modeling, if the plotted pressures are directed toward the inside of the enclosed domain, then adjust the pressure value with a negative sign.

This page intentionally left blank

6. Eigenvalue problems

ref. KJB
Section 6.8.2
and Chapter 10

The two types of analyses in ADINA for which the solution is obtained by solving the corresponding generalized eigenvalue problem are:

- ▶ Linearized buckling analysis
- ▶ Frequency analysis

These two types of analyses are discussed in the following sections.

6.1 Linearized buckling analysis

6.1.1 General considerations

The goal of linearized buckling analysis is to estimate a load level such that the tangent stiffness matrix corresponding to the load level has a zero eigenvalue.

A brief description of the theory of linearized buckling analysis is now given. Further information is given in the reference:

ref. K.J. Bathe and E.N. Dvorkin, "On the Automatic Solution of Nonlinear Finite Element Equations," *Computers & Structures*, Vol. 17, No. 5-6, pp. 871-879, 1983.

ref. KJB
Section 6.1

The solution in nonlinear static analysis is considered a function of time:

${}^t\mathbf{R}$ is the external load vector at time t ,

${}^t\mathbf{F}$ is the internal load vector at time t ,

and at equilibrium ${}^t\mathbf{R} - {}^t\mathbf{F} = \mathbf{0}$. The tangent stiffness matrix ${}^t\mathbf{K}$ is also considered a function of time.

The external load vector ${}^t\mathbf{R}$ is assumed to be a function of the load factor λ . The linearized buckling analysis solves for λ .

ADINA has two formulations for linearized buckling analysis: classical formulation and secant formulation. These formulations differ in how the external load vector ${}^t\mathbf{R}$ depends upon the load

factor λ .

Classical formulation)

For the classical formulation, we assume that the internal stress ${}^t\boldsymbol{\sigma}$ at each integration point scales with the load factor using

$${}^t\boldsymbol{\sigma} = \lambda \quad {}^{t_1}\boldsymbol{\sigma} \quad (6.1-1)$$

in which time t_1 is the time at the first time step. In other words, the internal stresses are assumed to be proportional to the load factor λ .

Assuming that the internal load vector is proportional to the stresses, and using that, at equilibrium, the external load vector equals the internal load vector,

$${}^t\mathbf{R} = \lambda \quad {}^{t_1}\mathbf{R} \quad (6.1-2)$$

We also assume that the stiffness matrix can be broken into linear and nonlinear parts:

$${}^t\mathbf{K} = {}^t\mathbf{K}_L + {}^t\mathbf{K}_{NL} \quad (6.1-3)$$

and furthermore that the linear part is not stress-dependent and the nonlinear part is proportional to the stresses ${}^t\boldsymbol{\sigma}$. This assumption is closely related to the construction of the stiffness matrix using the TL and UL formulations (see ref KJB, Section 6.2.3). With this assumption

$${}^t\mathbf{K} = {}^{t_1}\mathbf{K}_L + {}^t\lambda \quad {}^{t_1}\mathbf{K}_{NL} \quad (6.1-4)$$

Now we seek values of λ so that the determinant of ${}^t\mathbf{K}$, as constructed using (6.1-4), is equal to 0:

$$\det({}^{t_1}\mathbf{K}_L + \lambda_i \quad {}^{t_1}\mathbf{K}_{NL}) = 0 \quad (6.1-5)$$

Equation (6.1-5) can be rewritten using the definition $\gamma_i = \frac{\lambda_i - 1}{\lambda_i}$:

$$\det({}^t\mathbf{K} - \gamma_i({}^t\mathbf{K} - {}^t\mathbf{K}_{NL})) = 0 \quad (6.1-6)$$

Equation (6.1-6) corresponds to the eigenproblem

$${}^t\mathbf{K}\boldsymbol{\varphi}_i = \gamma_i({}^t\mathbf{K} - {}^t\mathbf{K}_{NL})\boldsymbol{\varphi}_i \quad (6.1-7)$$

where $\boldsymbol{\varphi}_i$ is the buckling mode shape for mode i .

It is clear from the above that the classical buckling formulation assumes equation (6.1-4), namely that the stiffness matrix can be decomposed into linear (stress-independent) and nonlinear (stress-dependent) parts. However, although equation (6.1-4) might be true for certain specialized problems, equation (6.1-4) is certainly not true in general. On the other hand, if ${}^t\mathbf{R}$ is chosen such that $\lambda_i \approx 1$, then $\gamma_i \approx 0$ and equation (6.1-6) implies that

$\det({}^t\mathbf{K}) \approx 0$, which means that the load level $\lambda_i \approx 1$ corresponds to buckling. Therefore, the closer ${}^t\mathbf{R}$ is to the actual buckling load, the more accurate the classical buckling formulation will be.

Secant formulation:

In the secant formulation, we assume

$${}^t\mathbf{R} = {}^{t_0}\mathbf{R} + \lambda({}^t\mathbf{R} - {}^{t_0}\mathbf{R}) \quad (6.1-8)$$

where time t_0 is the time at the start of the current analysis and time t_1 is the time at the first time step of the current analysis. (The current analysis only uses one time step.) When the current analysis is not a restart analysis, ${}^{t_0}\mathbf{R} = \mathbf{0}$, but when the current analysis is a restart analysis, ${}^{t_0}\mathbf{R} \neq \mathbf{0}$.

We also assume

$${}^t\mathbf{K} = {}^{t_0}\mathbf{K} + \lambda({}^t\mathbf{K} - {}^{t_0}\mathbf{K}) \quad (6.1-9)$$

Now we seek values of λ so that the determinant of ${}^t\mathbf{K}$, as constructed using (6.1-9), is equal to 0:

$$\det({}^{t_0}\mathbf{K} + \lambda_i({}^{t_1}\mathbf{K} - {}^{t_0}\mathbf{K})) = 0 \quad (6.1-10)$$

Again, substituting the definition $\gamma_i = \frac{\lambda_i - 1}{\lambda_i}$, equation (6.1-10) becomes

$$\det({}^{t_1}\mathbf{K} - \gamma_i {}^{t_0}\mathbf{K}) = 0 \quad (6.1-11)$$

which corresponds to the eigenproblem

$${}^{t_1}\mathbf{K}\boldsymbol{\phi}_i = \gamma_i {}^{t_0}\mathbf{K}\boldsymbol{\phi}_i \quad (6.1-12)$$

We see that the secant formulation is less restrictive than the classical formulation, since the secant formulation makes no assumptions about how the force vector and stiffness matrix depend on the internal stresses, and also the assumptions of equations (6.1-8) and (6.1-9) become better as t_0 and t_1 approach t .

As with the classical formulation, if ${}^{t_1}\mathbf{R}$ is chosen such that $\lambda_i \approx 1$, then $\gamma_i \approx 0$ and (6.1-11) implies that $\det({}^{t_1}\mathbf{K}) \approx 0$, which means that the load level at time t_1 corresponds to buckling, regardless of the choice of t_0 .

Comments

- The classical formulation is the default.
- Note that linearized buckling analysis must be specified in ADINA as a static analysis. In addition, only the subspace iteration method can be used in the eigenproblem solution.
- In the eigenvalue extraction, convergence of the eigenvalues γ_i is measured using

$$\frac{\gamma_i^{(k+1)} - \gamma_i^{(k)}}{\gamma_i^{(k+1)}} \leq \text{SSTOL}; \quad i = 1, 2, \dots, N$$

where N is the number of eigenmodes requested, k is the iteration counter and SSTOL is a user-specified convergence tolerance.

- If termination of solution occurs without attaining convergence, you should rerun the analysis using:

- (a) a larger load level for the first time step; and
- (b) a larger number of iteration vectors.

- A Sturm sequence check is always performed after the subspace iterations have converged to verify that indeed the smallest required eigenvalues and the corresponding mode shapes have been calculated. If this check fails, the program prints an approximation (λ_p) to the actual critical load factor (λ_1), where $\lambda_1 < \lambda_p$. In order to perform a successful analysis, you should rerun the analysis using (a) and (b) above.

- The eigenvalues are extracted in numerically ascending sequence. If the following eigenvalues are obtained from the eigenvalue solution:

$$\gamma_1 \leq \gamma_2 \leq \dots \leq \gamma_M < 1 < \gamma_{M+1} \leq \dots \leq \gamma_N$$

where N is the number of buckling modes requested, then the following inequalities hold for the critical load factors:

$$1 < \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_M$$

and

$$\lambda_{M+1} \leq \lambda_{M+2} \leq \dots \leq \lambda_N < 0$$

- The λ_1 value obtained should satisfy $1 < \lambda_1 < 500$. If this condition is violated, the program issues an error message and stops the execution, and you must adjust the applied load level for the first time step to successfully complete the analysis.
- At the completion of the eigenvalue solution, an error measure

for each eigenvalue and eigenvector is calculated using

$$\left\{ \begin{array}{c} \text{Error} \\ \text{measure} \\ \text{for} \\ \text{mode } i \end{array} \right\} = \left[\frac{(\mathbf{q}_i^{(k)})^T \mathbf{q}_i^{(k)}}{(\lambda_i^{(k)})^2} - 1 \right]^{\frac{1}{2}}$$

where $\mathbf{q}_i^{(k)}$ is the vector of the matrix \mathbf{Q}_k corresponding to $\lambda_i^{(k)}$ (see ref. KJB).

- With either the classical or secant formulations, the buckling load should be checked by increasing the load level until $\lambda_1 \approx 1$.
- Even with $\lambda_1 \approx 1$, the buckling load might not be correct if the stiffness matrix is not tangent. It is assumed throughout the linearized buckling analysis derivations that the stiffness matrix computed by ADINA is in fact a tangent stiffness matrix. It is also assumed that the solution tolerances are tight enough that equilibrium is in fact reached.

6.1.2 Introduction of geometric imperfections

- The buckling modes can be used as an initial geometry perturbation in a subsequent analysis. In a collapse analysis, the collapse loads can significantly depend on the geometric imperfections of the structure. A natural way to introduce appropriate geometric imperfections in a finite element analysis is to first perform a linearized buckling analysis and then superimpose on the "perfect" geometry of the finite element model a coordinate perturbation proportional to the buckling mode shapes. A collapse analysis can then be performed preferably using the LDC method on the perturbed model. See Section 11.16 for more details on geometric imperfections.
- The buckling loads should ideally be confirmed by imposing geometrical imperfections as described above, then running an LDC analysis. Verification problems B.43 and B.44, and primer problem 50, show the technique.

6.2 Frequency analysis

- The natural frequencies and mode shapes of vibration of the structural system are calculated using

$$\mathbf{K}\phi_i = \omega_i^2 \mathbf{M}\phi_i$$

where \mathbf{K} is the system stiffness matrix corresponding to the time of solution start, \mathbf{M} is the system mass matrix corresponding to the time of solution start, ω_i and ϕ_i are the angular frequency and mode shape, respectively, for mode i . Note that the frequencies are extracted in the eigenvalue solution in numerically ascending sequence. The eigenvectors are M-orthonormal, that is

$$\phi_i^T \mathbf{M} \phi_i = 1$$

- The effects of prescribed displacements are not included in \mathbf{K} .
- Frequencies and mode shapes can be computed in nonlinear analysis. The nonlinear analysis may contain geometric and material nonlinearities including contact, and the frequencies and mode shapes can be obtained at any solution step of the nonlinear analysis. The frequency solution corresponds to the eigensolutions of the tangent stiffness matrix and the mass matrix in the current equilibrium configuration. Physically, the frequencies and mode shapes in nonlinear analysis represent the response of the nonlinear structure to an infinitesimal perturbation about the equilibrium state. That is, it is the response when the nonlinear structure is “pinged” about its deformed configuration and allowed to undergo undamped vibrations.
- To obtain the frequencies and mode shapes of a nonlinear structure, the analysis must be performed in two runs. In the first run, the structure is loaded in a nonlinear static or dynamic analysis. This run can contain one or more solution steps. In the second run, you restart from the last solution of the first run.
- Frequencies and mode shapes can also be obtained in one nonlinear analysis run by using the ANALYSIS-SWITCH command. The program performs a step-by-step solution first, and

then automatically restarts at a given solution time to calculate frequencies and mode shapes.

- ANALYSIS-SWITCH allows multiple automatic restarts to frequency calculations from step-by-step solutions (statics, implicit dynamics or explicit dynamics).
- Modal stress analysis and/or modal participation factor calculations can also be performed when using ANALYSIS-SWITCH.
- The frequency analysis is based on the tangent stiffness matrix and mass matrix at the last converged state. Therefore, the stiffness and mass matrices include all geometric and material nonlinearities corresponding to the end of the nonlinear static or dynamic analysis.
- When a frequency analysis is performed on a nonlinear structure, only those effects that are included in the tangent stiffness matrix are accounted for, such as:
 - ▶ Geometric nonlinear stiffening effects, e.g. stress stiffening and centrifugal spin softening.
 - ▶ Contact.
 - ▶ Mesh glue.
- The following effects are not included in the tangent stiffness matrix, and hence are not accounted for in the frequency analysis:
 - ▶ Stiffness contribution due to deformation-dependent pressure and concentrated follower force loading for large displacement analysis.
 - ▶ Spin softening effects and the damping terms due to Coriolis loading. This requires a non-symmetric complex eigen-solver which is not supported in ADINA.
 - ▶ Prescribed displacements. A stiff spring (1E30 times stiffer than any DOF in the assemblage) with a load to deform the spring by the prescribed displacement amount should be

used instead.

- When contact is included in frequency analysis, certain contactor node degrees of freedom are assumed fixed (glued) to the target segment:

For frictionless contact:

- Contactor nodes whose contact status is OPEN are assumed free.
- Contactor nodes whose contact status is CLOSED have their normal DOF fixed (glued) to the target segment in frequency analysis. The tangential DOF are assumed free (slipping contact).
-

For frictional contact:

- Contactor nodes whose contact status is OPEN are assumed free.
 - Contactor nodes whose contact status is STICKING (sliding velocity less than or equal to EPST) have all their DOF fixed to the target segment in frequency analysis. In this case, the contactor node is assumed glued to the target segment.
 - Contactor nodes whose contact status is SLIPPING (sliding velocity greater than EPST) only have their normal DOF fixed to the target segment in frequency analysis. The tangential DOF are assumed free (slipping contact).
- In frictional contact frequency analysis, if all contactor nodes should be fixed, the load should be held constant for several time steps until the contact status is STICKING for all contactor nodes. Recall that EPST is inversely proportional to the relaxation time for frictional contact. Hence, the larger EPST, the quicker the contact status will change to STICKING.
 - The CONT-COMPL parameter of the FREQUENCIES command specifies whether the effect of contact compliance is accounted for in frequency analysis. By default, the effect of

contact compliance is accounted for.

Contact-compliance can significantly affect the frequencies and mode shapes, as it artificially softens the contact surface.

When obtaining predictions for the frequencies and mode shapes of the actual contacting bodies, the effect of contact-compliance should not be included in the frequency analysis.

Including the effect of contact-compliance in frequency analysis can be useful to determine whether an oscillation observed in the nonlinear dynamic analysis solution is due to the excitation of a contact-compliance “bouncing” mode.

- At the completion of the eigenvalue solution, an error measure for each eigenvalue and eigenvector is calculated using

$$\left\{ \begin{array}{c} \text{Error} \\ \text{measure} \\ \text{for} \\ \text{mode } i \end{array} \right\} = \left[\frac{(\mathbf{q}_i^{(k)})^T \mathbf{q}_i^{(k)}}{(\lambda_i^{(k)})^2} - 1 \right]^{\frac{1}{2}}$$

where $\mathbf{q}_i^{(k)}$ is the vector of the matrix \mathbf{Q}_k corresponding to $\lambda_i^{(k)}$ (see ref. KJB).

- If there are frequencies that are numerically zero, the eigensolution procedures can still calculate the frequencies and corresponding mode shapes. Frequencies that are numerically zero are present in the model if all rigid body modes have not been suppressed, or if there are input or modeling errors. The number of rigid body modes in the model is equal to the number of zero frequencies.

In order to obtain a frequency solution when there are zero frequencies, you must activate the "allow rigid body modes" option, otherwise ADINA will stop. This option is also useful if any part of the model would be insufficiently supported (i.e., have rigid body modes) if all contact, mesh glueing and generalized constraints are removed.

When the rigid body mode shift option is used, you can enter the rigid body mode shift value (which must be negative). If you instead use the default shift value of 0.0, ADINA calculates it using the formula

$$\text{RBMSH} = f \min_{i=1,\dots,N} \left[\frac{k_{ii}}{m_{ii}} \right]$$

where $f = -10^{-3}$ for the determinant search method or if contact is present, $f = -10^{-7}$ otherwise and N is the number of mass degrees of freedom in the finite element model.

- The frequencies and mode shapes of vibration can be calculated using the following methods: **determinant search**, **subspace iteration** or **Lanczos iteration**.
- Frequency analysis including potential-based fluid elements can be performed using only the determinant search or Lanczos iteration methods. In addition, the frequency analysis cannot have potential-based fluid elements together with contact, glueing, or generalized constraints.

6.2.1 Determinant search method

*ref. KJB
Section 11.4*

- The use of the determinant search method is usually effective when
 - ▶ A narrow banded finite element system is considered.
 - ▶ The total stiffness matrix can be processed in-core.
- The determinant search method calculates the smallest frequencies and corresponding mode shapes of the system.

6.2.2 Bathe subspace iteration method

*ref. KJB
Section 11.6*

- The Bathe subspace iteration method used in ADINA contains all the capabilities of the original basic subspace iteration method described in

ref. K.J. Bathe, "Solution Methods of Large Generalized Eigenvalue Problems in Structural Engineering," *Report UC SESM 71-20*, Univ. of California, Berkeley, 1971.

plus the acceleration techniques developed more recently:

ref. K.J. Bathe and S. Ramaswamy, "An Accelerated Subspace Iteration Method," *J. Computer Meth. in Applied Mech. and Eng.*, Vol. 23, pp. 313-331, 1980.

- When the acceleration techniques are employed, the method can be significantly more effective. These techniques consist of

- ▶ The use of the Lanczos method to generate the starting subspace
- ▶ Vector over-relaxation in the vector iterations

- The default number of iteration vectors, q , used is $\max\{2p, p+8\}$ when p frequencies and associated mode shapes are to be calculated. For experiences with the method see:

ref. K. J. Bathe, "The Subspace Iteration Method — Revisited", *Computers & Structures*,
j.compstruct.2012.06.02

- The subspace iteration can be employed to calculate the smallest frequencies and corresponding mode shapes, or the frequencies (and mode shapes) within a certain prescribed interval.

- If q iteration vectors are used and there are only q mass degrees of freedom, the subspace iteration method corresponds to a static condensation (Guyan reduction) and subsequent solution of the eigenvalues.

This analysis feature may be useful when you assign zero mass density to the material, and account for the mass by means of concentrated masses corresponding to a few degrees of freedom only. However, note that such mass lumping requires much experience with the problem to be solved and the finite element model used.

ref. KJB
Sections 10.3.1
10.3.2 and
11.6.3

- The standard Lanczos method calculates only approximations to the eigenvalues and eigenvectors, and for this reason it is here only used to generate the starting subspace. If the Lanczos method is employed, it is usually effective to use a number of iteration vectors q considerably larger than the number of frequencies

ref. KJB
Section 11.5

required p , i.e., use $q = 2p$, with q still much smaller than the total number of frequencies of the finite element system.

- ADINA can perform a Sturm sequence check after the subspace iterations have converged to verify that the smallest required eigenvalues and corresponding mode shapes have been calculated. The cost of the Sturm sequence check is approximately equal to the cost of a static solution.
- The subspace iteration method can be employed to calculate frequencies within an interval. A lower and upper bound of frequencies need to be specified, and the maximum number of frequencies ADINA can calculate. The program then checks how many frequencies exist in the specified interval and calculates them all if the number of existing frequencies is smaller or equal to the maximum number of frequencies specified. If the number of existing frequencies is larger than the specified maximum number of frequencies, the program stops with a message that more frequencies exist than the maximum number provided via input.

6.2.2.1 Enriched Bathe subspace iteration method

- The enriched Bathe subspace iteration method is the implementation of a novel extension of the Bathe subspace iteration method for the solution of generalized eigenvalue problems in structural dynamics. The method is presented in the following paper:

ref. K.T. Kim and K.J. Bathe, “The Bathe subspace iteration method enriched by turning vectors”, *Computers & Structures*, 186 (2017), 11-21

- As a first step, the Method uses the standard (Bathe) subspace iteration method to perform one iteration using standard starting vectors.
- The Method then uses a 10-step procedure described in the above reference, where turning vectors are calculated to achieve fast convergence.

- After convergence is reached, a Sturm sequence check can be performed.
- The default number of iteration vectors, q , used is $\max(1.4p, p+8)$, where p frequencies and associated mode shapes are to be calculated.
- ADINA can perform a Sturm sequence check after the Lanczos iterations have converged to verify that the smallest required eigenvalues and corresponding mode shapes have been calculated.

6.2.3 Lanczos iteration method

*ref. KJB
Section 11.5*

- The Lanczos method can be used to calculate the smallest frequencies and corresponding mode shapes. The method can be very effective for large problems.
- When potential-based fluid elements are present, the Lanczos method should be used. (For small problems, the determinant search method can also be used; for large problems, the determinant search method is ineffective.)
- The convergence of the iteration is assessed by the criterion

$$\frac{\gamma_i^{(k-1)} - \gamma_i^{(k)}}{\gamma_i^{(k-1)}} \leq \varepsilon; \quad i = 1, \dots, N$$

where N is the number of eigenmodes requested, k is the Lanczos iteration stage counter and the value $\varepsilon = 10^{-9}$ is used. See ref. KJB, Section 11.5 for all notation not described here.

- After convergence, the error bound $|\beta_q s_{qi}|$ and the physical error estimate $\frac{\|\mathbf{K}\phi_i - \omega_i^2 \mathbf{M}\phi_i\|}{\|\mathbf{K}\phi_i\|}$ are calculated.

Since the calculated vectors in the Lanczos method may, in rare cases, lose orthogonality, when ADINA performs a Sturm sequence

check, ADINA also calculates the M-orthogonal checks

$$DII = \max_{1 \leq i \leq n} \left| \left(\phi_i^T \right)^{t_0} \mathbf{M}(\phi_i) - 1 \right|, \quad DIJ = \max_{\substack{1 \leq i, j \leq n \\ i \neq j}} \left| \left(\phi_i^T \right)^{t_0} \mathbf{M}(\phi_j) \right|$$

to verify that the eigenvectors are M-orthogonal.

- This method is the default technique used in ADINA
- For potential-based fluid elements, when a relatively large number of frequencies needs to be calculated, it can be effective, and sometimes necessary, to use the shifting procedure. The option to use the shifting procedure is controlled by the parameters `NSHIFT` and `NSHIFT-BLOCK` in the `FREQUENCIES` command.
- ADINA can perform a Sturm sequence check after the Lanczos iterations have converged to verify that the smallest required eigenvalues and corresponding mode shapes have been calculated. For models with potential-based fluid elements, the Sturm sequence check is always performed.
- For potential-based elements, the error measures printed in the output file should always be checked to make sure that they are not large. The use of parameter `RSHIFT` in the `FREQUENCIES` command can sometimes be useful to obtain solutions with small physical error norms, for example `RSHIFT=-0.005`.
- The Lanczos method cannot be used in conjunction with potential-based fluid elements and any of the following: contact, glueing or generalized constraints.

6.2.4 Modal stresses

- Modal strains, stresses, forces and reactions can be requested in ADINA. To calculate these quantities, small displacements and small strains are assumed. Modal displacements and modal stresses are therefore proportional.
- A frequency analysis must be performed first to obtain the modal displacements. This analysis can follow a materially or

geometrically nonlinear analysis.

- In linear analysis, the modal stresses are directly calculated from the **M**-orthonormal modal displacements ϕ_i , where

$$\phi_i^T \mathbf{M} \phi_i = 1.$$

- In materially or geometrically nonlinear analysis, the modal stresses for mode i ($1 \leq i \leq N$) are calculated using the approximation

$$\sigma_i^m = \frac{(\sigma_i(\mathbf{U}_i^m) - {}^t\sigma)}{\alpha_i}$$

where $\sigma_i(\mathbf{U}_i^m)$ are the stresses computed from the perturbed displacement vector $\mathbf{U}_i^m = {}^t\mathbf{U} + \alpha_i \phi_i$, ${}^t\sigma$ is the stress vector at time t , and α_i is a perturbation factor, controlled by parameter DUSIZE as follows.

If DUSIZE is not equal to 0, then $\alpha_i = \frac{\text{DUSIZE}}{\|\phi_i\|}$; this choice

implies $\|\mathbf{U}_i^m - {}^t\mathbf{U}\| = \text{DUSIZE}$. Here the symbol $\|\cdot\|$ is the norm of the enclosed vector only including translations. DUSIZE is seen to be the size of the displacement perturbation.

If DUSIZE is equal to 0, then ADINA computes

$$\alpha_i = 0.001 \frac{\|{}^t\mathbf{U}\|}{\|\phi_i\|} \text{ where } {}^t\mathbf{U} \text{ is the total displacement vector at time } t$$

(last step of analysis before frequency analysis). This choice implies $\|\mathbf{U}_i^m - {}^t\mathbf{U}\| = 0.001 \|\mathbf{U}\|$, which is usually quite reasonable. In the special case in which the total displacement vector is zero (for example, if the modal stress calculation is requested at the start of the analysis), then ADINA sets α_i to 0.001.

The choice $\alpha_i = 0.001$ can lead to very large or very small displacement perturbations because then $\|\mathbf{U}_i^m - {}^t\mathbf{U}\| = \frac{0.001}{\|\phi_i\|}$ and, depending upon the units used, $\|\phi_i\|$ can be very large or very

small. So we recommend that you specify DUSIZE when the total displacement vector is zero.

This page intentionally left blank

7. Static and implicit dynamic analysis

This chapter presents the formulations and algorithms used to solve static and dynamic problems using ADINA. This includes convergence checking and the available solvers.

Information about the progress of the solution are always output to the .out file. A shorter summarized output is provided in the .msg file or the dialog box.

7.1 Linear static analysis

*ref. KJB
Sections 8.2.1,
8.2.2 and 8.2.3*

- In linear analysis using ADINA, the finite element system equilibrium equations to be solved are

$$\mathbf{KU} = \mathbf{R}$$

- For the solution of these equations, the following solvers are available: sparse solver, iterative solver, iterative multigrid solver, 3D-iterative solver. The sparse solver is the default.
- The equation solvers assume that the system stiffness matrix is symmetric.
- The equation solvers assume that the system stiffness matrix is positive definite. This requirement can be summarized as follows: The Rayleigh quotient

$$\rho(\phi) = \frac{\phi^T K \phi}{\phi^T \phi}$$

must be greater than zero for any displacement vector ϕ . Since $\rho(\phi)$ is equal to twice the strain energy stored in the system (for $\phi^T \phi = 1$), this is equivalent to the requirement that the strain energy stored in the finite element system when subjected to any displacement vector ϕ must be greater than zero.

- Hence, the finite element system must be properly supported,

so that the system cannot undergo any rigid body displacements or rotations.

- It also follows that no part of the total finite element system must represent a mechanism, see ref. KJB, Fig. 8.7, p. 704, for such a case.

- Nodal point degrees of freedom for which there is no stiffness must have been deleted. A degree of freedom does not carry any stiffness if all of the elements connected to the nodal point do not carry stiffness into that degree of freedom. In this case the degree of freedom must be deleted using any one of the boundary conditions.

Note that nodal degrees of freedom which are not connected to any elements and are not used as slave nodes in constraint equations are (by default) automatically deleted by the AUI.

- The elements joining into a nodal point and contributing to the stiffness of the nodal point degrees of freedom must all be defined properly and in a physically correct manner. For example, if there are program input errors that yield a zero or negative Young's modulus or an incorrect nodal point numbering for an element, the stiffness at a system degree of freedom may be zero or negative.

7.1.1 Sparse solver

- The main direct solution method in ADINA is a sparse matrix solver.
- The ADINA sparse matrix solver has been proven to yield drastic reductions in solution times, being one to two orders of magnitude faster than the active column solver. In addition, disk and overall memory requirements are reduced.
- The ADINA sparse matrix solver is very reliable and robust and is generally to be used for most problems in ADINA. The ADINA sparse matrix solver is the default solver.
- The sparse solver allocates memory outside that allocated by ADINA. Therefore the total memory used is the sum of the

memory reserved by ADINA plus that dynamically allocated by the sparse solver code.

- The sparse solver can be used both in-core and out-of-core. It should be noted that it is more efficient to run an out-of-core sparse solver using real (physical) memory than it is to run an in-core sparse solver using virtual memory. Therefore, if real memory is limited, we recommend the following guidelines:

1) The memory allocation MTOT for ADINA should be specified just high enough to allow ADINA to use IOPTIM=3 (refer to the output file for the printout of the IOPTIM value).

2) In the AUI, you should provide a memory limit for the sparse solver using the following formula

$$\text{MAXSOLMEM} \leq 0.85 \times (\text{real memory of your computer}) - \text{MTOT}$$

in which MAXSOLMEM, (real memory) and MTOT are measured in MW (or MB).

- When a non-positive definite stiffness matrix (i.e. one with a zero or negative diagonal element) is encountered during solution, ADINA may stop or continue, according to the following rules:
 - ▶ If a diagonal element is exactly equal to 0.0, ADINA stops unless
 - The equation number corresponding to the zero diagonal element is attached to an inactive element (element birth/death option is used).
 - You have requested that ADINA continue execution (see below).
 - ▶ If the value of a diagonal element is smaller than 10^{-12} but not equal to zero, or the value of a diagonal element is negative, ADINA stops unless one of the following options is used:
 - Automatic load-displacement (LDC)

- Automatic time-stepping (ATS)
- Potential-based fluid elements
- Contact analysis

or unless you have requested that ADINA continue execution.

- When ADINA stops, ADINA prints informational messages for the zero or negative diagonal elements, up to 25% of the number of equations.
- When ADINA continues execution, ADINA assigns a very large number to the diagonal element, effectively attaching a very stiff spring to that degree of freedom.
- You can request that ADINA continue execution using the control parameter “Continue Even When Non-Positive Definite Stiffness Matrix Encountered” in the AUI Control→Solution Process dialog box.
- Note that the stiffness matrix can be non-positive definite due to a modeling error, for example if the model is not sufficiently restrained in static analysis. In this case the results obtained can be misleading.

7.1.2 Iterative and multigrid solvers

- In the analysis of large problems, the amount of storage required by a direct solution method may be too large for the available computer resources. For such problems, the use of the iterative method of solution is necessary.
- Iterative solution methods related to those used in ADINA are described in ref. KJB, Section 8.3.2.
- The iterative solver can be used with all solution options of ADINA, in linear and nonlinear, static and dynamic analysis (except for frequency and linearized buckling analysis). The conditioning of the coefficient matrix is generally significantly better in dynamic analysis because of the effects of the mass matrix (inertia effects) in the coefficient matrix.

- The iterative solver cannot recognize that the stiffness matrix has a zero pivot. If the stiffness matrix is not positive definite, the iteration will continue without reaching convergence.

- The main practical differences between the use of the direct solver and the iterative solver are as follows:

- ▶ The direct solver executes a predetermined number of operations after which the solution is obtained.

- ▶ The iterative solver performs a predetermined number of operations per iteration, but the number of iterations is not known beforehand. The number of iterations depends on the condition number of the coefficient matrix: the higher the condition number, the more iterations are needed.

The number of iterations required varies from a few hundred to a few thousand. In general, the solution of bulky 3-D solids requires significantly less iteration than the solution of thin structural configurations.

- Regarding the convergence of the iterative method, assume that the system of equations to be solved is $\mathbf{Ax} = \mathbf{b}$, where $\mathbf{x}^{(k)}$ is the approximate solution at iteration k , and the residual vector is $\mathbf{r}^{(k)} = \mathbf{b} - \mathbf{Ax}^{(k)}$. The iterative method is converged when either

$$\|\mathbf{r}^{(k)}\|_2 \leq \text{EPSIA} \quad \text{and} \quad \frac{\|\mathbf{r}^{(k)}\|_2}{\|\mathbf{b}\|_2} \leq \text{EPSIB}$$

or

$$\|\mathbf{r}^{(k)}\|_2 \leq \text{EPSII}$$

Note that EPSII should be less than or equal to EPSIA. The defaults are EPSIA=1E-6, EPSIB=1E-4, EPSII=1E-8. However, for nonlinear analysis with equilibrium iterations, looser tolerances

can be used, e.g. EPSIA=1E-4, EPSIB=1E-3, EPSII=1E-5.

- Difficulties might be encountered when the matrix **A** is ill-conditioned. For example, static shell problems are difficult to solve because shell structures are much stiffer in membrane action than in bending action. For such cases, a shift factor SHIFT can be effective, with $\text{SHIFT} > 1.0$, e.g. $\text{SHIFT}=1.02$. SHIFT is used to scale the off-diagonal entries of **A** by $1/\text{SHIFT}$ in order to make the preconditioner more diagonally dominant.
- ADINA also includes a multigrid solver. The multigrid solver is intended for use with very large systems of equations. The multigrid solver is efficiently used when the model consists of 3-D tetrahedral solid elements that were generated using free-form meshing techniques.

7.1.3 3D-iterative solver

- The 3D-iterative solver has been developed to efficiently solve large models (i.e. problems with more than 500,000 equations) containing mainly higher order 3-D solid elements.
- In addition to the higher order 3-D solid elements, the models can contain other elements available in the program (e.g., shells, rods, beams, rebars, etc.), including contact conditions.
- The 3D-iterative solver is effective in linear or nonlinear static analysis and in nonlinear dynamic analysis. For linear dynamic analysis, the sparse solver is usually more effective.
- The 3D-iterative solver, like all iterative solvers, performs a number of iterations until convergence is reached.
- In linear analysis, the 3D-iterative solver convergence tolerance affects the accuracy of the solution. If the convergence tolerance (EPSN parameter of the SOLVER ITERATIVE command) is too loose, inaccurate results are obtained, and if the tolerance is too tight, much computational effort is spent to obtain needless accuracy.
- In nonlinear analysis, the 3D-iterative solver is used to solve the linearized equations in each Newton-Raphson equilibrium iteration.

Hence, in this case, the 3D-iterative solver convergence tolerance only affects the convergence rate, not the accuracy of the solution. The accuracy of the solution is specified by the equilibrium iteration tolerances (using the TOLERANCES ITERATION command).

- The benefits of the 3D-iterative solver are:
 - ▶ When the 3D-iterative solver is used, the solution time and memory requirements scale approximately linearly with the number of equations. When the sparse solver is used, the solution time and memory requirements scale approximately quadratically with the number of equations. Therefore, the 3D-iterative solver allows the solution of very large problems.
 - ▶ The 3D-iterative solver is significantly more stable than the multigrid iterative solver; hence, the 3D-iterative solver should always be used.
- The limitations of the 3D-iterative solver are:
 - ▶ The 3D-iterative solver convergence tolerance (EPSN) must be appropriately set.
 - ▶ The 3D-iterative solver does not scale well for shared memory parallel processing. However, the sparse matrix solver scales well to about 8-cores.
 - ▶ Nearly incompressible hyperelastic materials may slow down the convergence of the 3D-iterative solver. For these material models, the bulk modulus κ should be restricted to a value corresponding to $\nu = 0.49$, instead of the default 0.499 (see Eq. 3.8-13)
- Convergence control in the 3D-iterative solver is as follows:

Considering the linearized equation $Ax = b$, let x^i be its approximate solution at inner-iteration i and $r^i (= b - Ax^i)$ be its corresponding residual. The convergence in the iterative solver is said obtained if any one of the following criteria is satisfied

$$\|b\| \leq \varepsilon \quad \text{or} \quad \|r^i\| \leq \varepsilon \quad \text{or} \quad \|x^i - x^{i-1}\| \leq \varepsilon$$

$$\text{or equation residual:} \quad \|r^i\|/r_{scale} \leq \begin{cases} 0.001\varepsilon_b & i < 3 \\ \varepsilon_b & i \geq 3 \end{cases}$$

$$\text{or solution residual:} \quad \|x^i - x^{i-1}\|/x_{scale} \leq \begin{cases} 0.001\sigma_v & i < 3 \\ \sigma_v & i \geq 3 \end{cases}$$

$$\text{The solution norm is} \quad \|x\| = \frac{1}{n} \sum_{i=1}^n |x_i|$$

In the above, $\varepsilon \equiv 10^{-16}$, the equation scale $r_{scale} = \|b\|$, and the variable scale

$$x_{scale} = \begin{cases} \max \left\{ \varepsilon, \frac{1}{3} (\|x^1\| + \|x^2\| + \|x^3\|) \right\} & \text{for linear problems} \\ \max \left\{ \varepsilon, \frac{1}{3} (\|x^1\| + \|x^2\| + \|x^3\|), \|\Delta x\| \right\} & \text{for nonlinear problems} \end{cases}$$

where $\Delta x = {}^{t+\Delta t}U - {}^tU$, $x^i = \Delta U^{(i)}$ is the current solution increment in the Newton-Raphson iteration i , $\varepsilon_b = \text{EPSIB}$ of the SOLVER ITERATIVE command, and $\sigma_v = \min(10^{-6}, \varepsilon_b \cdot 10^{-3})$.

The equation residual (EQ) and the solution residual (VAR) are available in the .out file.

7.2 Nonlinear static analysis

ref. KJB
Section 8.4

- In nonlinear static analysis the equilibrium equations to be solved are:

$${}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F} = \mathbf{0}$$

where ${}^{t+\Delta t}\mathbf{R}$ is the vector of externally applied nodal loads at time (load) step $t+\Delta t$, and ${}^{t+\Delta t}\mathbf{F}$ is the force vector equivalent (in the virtual work sense) to the element stresses at time $t+\Delta t$.

- The nonlinearity may come from the material properties, the

kinematic assumptions, the use of contact surfaces, or the use of special features such as the element birth/death option.

- The solution to the static equilibrium equations can be obtained in ADINA using
 - ▶ Load/displacement incrementation without iteration
 - ▶ Modified Newton iterations, with or without line searches
 - ▶ The BFGS (Broyden-Fletcher-Goldfarb-Shanno) method
 - ▶ Full Newton iterations, with or without line searches
 - ▶ Automatic step incrementation (automatic-time-stepping and load-displacement-control methods)

These methods are described in detail in the following sections and also in Sections 6.1 and 8.4 of ref. KJB.

- The same equation solvers are used in nonlinear analysis as in linear analysis. In addition, a non-symmetric sparse solver is available.
- The automatic time stepping algorithms do not require the stiffness matrix to be positive definite, thus allowing for the solving of bifurcation problems.
- A special case is reached when the element birth or death option is used for some elements. In this case the equation solver allows that during the solution a degree of freedom may not carry any stiffness at a particular time, because the adjoining element(s) may not have been born yet (or may have died already). Note that the program does not check if any load is applied in that degree of freedom, and that the appropriate time functions should be used to generate/delete that load according to the element birth or death.
- The stiffness stabilization feature can be used to treat some nonlinear static problems involving a non-positive definite stiffness matrix. See section 11.14 for details.

7.2.1 Solution of incremental nonlinear static equations

Load/displacement incrementation without iteration: The following algorithm is used if equilibrium iteration is not performed:

$$\begin{aligned}\tau\mathbf{K}\mathbf{U} &= {}^{t+\Delta t}\mathbf{R} - {}^t\mathbf{F} \\ {}^{t+\Delta t}\mathbf{U} &= {}^t\mathbf{U} + \mathbf{U}\end{aligned}$$

where $\tau\mathbf{K}$ is the tangent stiffness matrix at time τ , where $\tau \leq t$; ${}^{t+\Delta t}\mathbf{R}$ is the externally applied load vector at time $t+\Delta t$; ${}^t\mathbf{F}$ is the consistent nodal force vector corresponding to the element stresses due to the displacement vector ${}^t\mathbf{U}$, and \mathbf{U} is the incremental displacement vector.

Note that $\tau\mathbf{K}$ is the stiffness matrix corresponding to the last stiffness reformation. Stiffness reformations are performed only at the solution steps that you specify.

Since equilibrium iteration is not performed, the solution in general does not satisfy accurately nodal point equilibrium. Hence, in practice this method should only be used with a sufficiently small time step increment so that divergence from the equilibrium solution is small. Whether nodal point equilibrium is sufficiently satisfied can be approximately checked by performing equilibrium iterations at the final solution step and noting how many iterations are required.

Modified Newton iteration: In modified Newton iteration the following algorithms are used:

- Without line search

$$\begin{aligned}\tau\mathbf{K}\Delta\mathbf{U}^{(i)} &= {}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F}^{(i-1)} \\ {}^{t+\Delta t}\mathbf{U}^{(i)} &= {}^{t+\Delta t}\mathbf{U}^{(i-1)} + \Delta\mathbf{U}^{(i)}\end{aligned}$$

(7.2-1a,b)

- With line search

$$\begin{aligned} {}^{\tau}\mathbf{K} \Delta \mathbf{U}^{(i)} &= {}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F}^{(i-1)} \\ {}^{t+\Delta t}\mathbf{U}^{(i)} &= {}^{t+\Delta t}\mathbf{U}^{(i-1)} + \beta^{(i)} \Delta \mathbf{U}^{(i)} \end{aligned} \quad (7.2-1c,d)$$

where ${}^{\tau}\mathbf{K}$ is the tangent stiffness matrix at time τ , where $\tau \leq t$; ${}^{t+\Delta t}\mathbf{R}$ is the externally applied load vector at time $t + \Delta t$; ${}^{t+\Delta t}\mathbf{F}^{(i-1)}$ is the consistent nodal force vector corresponding to the element stresses due to the displacement vector ${}^{t+\Delta t}\mathbf{U}^{(i-1)}$; $\Delta \mathbf{U}^{(i)}$ is the incremental displacement vector in iteration (i) and $\beta^{(i)}$ is an acceleration factor obtained from line search.

Note that ${}^{\tau}\mathbf{K}$ is the stiffness matrix corresponding to the last stiffness reformation. Stiffness reformations are performed only at the solution steps that you specify.

Modified Newton iterations are performed only at the solution steps that you specify.

ref. KJB
Section 8.4.2

BFGS method: In the BFGS matrix update method, the following algorithm is used:

$$\begin{aligned} {}^{t+\Delta t}\mathbf{K}^{*(i-1)} \Delta \mathbf{U}^{(i)} &= {}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F}^{(i-1)} \\ {}^{t+\Delta t}\mathbf{U}^{(i)} &= {}^{t+\Delta t}\mathbf{U}^{(i-1)} + \beta^{(i)} \Delta \mathbf{U}^{(i)} \end{aligned}$$

where ${}^{t+\Delta t}\mathbf{K}^{*(i-1)}$ is an updated stiffness matrix (based on the iteration history). The definitions of the other variables are as for Eq. (7.2-1).

Note that ${}^{t+\Delta t}\mathbf{K}^{*(i-1)}$ is not explicitly formed, but instead the inverse of the stiffness matrix is updated using vector products to provide a secant approximation to the stiffness matrix in successive iterations.

BFGS matrix updates are performed only at the solution steps that you specify.

Full Newton iterations: In the full Newton iteration method, the algorithms employed without and with line searches are:

- Without line search

$${}^{t+\Delta t}\mathbf{K}^{(i-1)} \Delta \mathbf{U}^{(i)} = {}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F}^{(i-1)}$$

$${}^{t+\Delta t}\mathbf{U}^{(i)} = {}^{t+\Delta t}\mathbf{U}^{(i-1)} + \Delta \mathbf{U}^{(i)}$$

- With line search

$${}^{t+\Delta t}\mathbf{K}^{(i-1)} \Delta \mathbf{U}^{(i)} = {}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F}^{(i-1)}$$

$${}^{t+\Delta t}\mathbf{U}^{(i)} = {}^{t+\Delta t}\mathbf{U}^{(i-1)} + \beta^{(i)} \Delta \mathbf{U}^{(i)}$$

where ${}^{t+\Delta t}\mathbf{K}^{(i-1)}$ is the tangent stiffness matrix based on the solution calculated at the end of iteration $(i - 1)$ at time $t + \Delta t$.

The definitions of the other variables are as for Eq. (7.2-1). Note that when the full Newton iteration method is employed a new stiffness matrix is always formed at the beginning of each new load step and in each iteration.

- An upper bound for the incremental displacements in $\Delta \mathbf{U}$ can be set by the user. If the largest increment displacement component exceeds the limiting value, the whole incremental displacement vector is scaled down to satisfy the upper bound.

This feature is useful for problems where one or more iterations can produce unrealistically large incremental displacements. This may happen, for example, if a load is applied to contacting bodies before contact is properly established, or in the first unloading steps after a material has undergone plastic deformation.

- The minimum number of equilibrium iterations per solution step can be specified by the user. The default minimum number of equilibrium iterations is 1 for problems without contact and is 2 for problems with contact.
- Nodal and element results can be saved in the porthole (.por) file during equilibrium iterations by using the RESULTS-ITE command. Iteration results are included in the standard porthole file, which can be loaded into the AUI for post-processing. Two options are available: iteration results can be saved in the porthole file due to non-convergence, or for a given solution time.

7.2.2 Line search

When the line search feature is activated, the incremental displacements obtained from the solver are modified as follows

$${}^{t+\Delta t}\mathbf{U}^{(i)} = {}^{t+\Delta t}\mathbf{U}^{(i-1)} + \beta^{(i)}\Delta\mathbf{U}^{(i)}$$

where β is a scaling factor obtained from a line search in the direction $\Delta\mathbf{U}^{(i)}$ in order to reduce out-of-balance residuals, according to the following criterion

$$\frac{\Delta\mathbf{U}^{(i)T} \left[{}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F}^{(i)} \right]}{\Delta\mathbf{U}^{(i)T} \left[{}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F}^{(i-1)} \right]} \leq \text{STOL} \quad (7.2-2)$$

where STOL is a user-input line search convergence tolerance, and ${}^{t+\Delta t}\mathbf{F}^{(i)}$ is calculated using the total displacement vector ${}^{t+\Delta t}\mathbf{U}^{(i)}$.

The magnitude of β is also governed by the following bounds

$$\text{LSLOWER} \leq \beta \leq \text{LSUPPER} \quad (7.2-3)$$

where LSUPPER and LSLOWER are user-input parameters.

The incremental displacements are not modified (i.e., $\beta = 1$) if no suitable line search parameter satisfying Equations (7.2-2) and (7.2-3) is found within a reasonable number of line search iterations, or if the unbalanced energy falls below a certain user-specified energy threshold ENLSTH.

- Line search is off by default except for the BFGS method. It is useful for problems involving plasticity, as well as large displacement problems involving beams and shells. It is also helpful in many contact problems. In the case of contact problems, it is sometimes better to set LSUPPER to 1.0 so that the line search only scales down displacements.
- The effect of line search is more prominent when the current displacements are far from the converged solution. This usually happens in the first few iterations of a time step, or when a major

change occurs in the model, due for example, to contact initiation/separation, or the onset of plasticity.

- Note that line search increases the computational time for each iteration. Most of the extra time goes towards the evaluation of ${}^{t+\Delta t}\mathbf{F}^{(i)}$ in Equation (7.2-2). However, for the types of problems mentioned above the reduction in the number of iterations and the ability to use bigger time steps leads to an overall reduction in solution time.

7.2.3 Low speed dynamics feature

- A low speed dynamics option is available for static analysis (can only be used with ATS or TLA/TLA-S features). Low speed dynamics is a special technique developed to overcome convergence difficulties in collapse, post-collapse and certain contact problems.

In essence, this feature includes dynamics effects in an otherwise static problem. ADINA solves

$$\delta \mathbf{M} {}^{t+\Delta t}\ddot{\mathbf{U}}^{(i)} + \mathbf{C} {}^{t+\Delta t}\dot{\mathbf{U}}^{(i)} + {}^{t+\Delta t}\mathbf{K}^{(i-1)} \Delta \mathbf{U}^{(i)} = {}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F}^{(i-1)} \quad (7.2-4)$$

where \mathbf{M} is the mass matrix and δ is a mass scaling factor that can vary from 0.0 to 1.0 (default 1.0), to partially account for the dynamic inertia effect. The \mathbf{C} matrix is evaluated using

$$\mathbf{C} = \beta \mathbf{K}$$

where β is a user-specified parameter (default 10^{-4}) and \mathbf{K} is the (initial) total stiffness matrix (corresponding to zero initial displacements). For details on this dynamic equation refer to Section 7.4.

- If the METHOD=BACKWARD (Euler-backward) option is chosen for low speed dynamics in the AUTOMATIC TIME-STEPPING command, then the following governing equation is used to model a semi-static response:

$$\mathbf{C} {}^{t+\Delta t}\dot{\mathbf{U}} + {}^{t+\Delta t}\mathbf{K}^{(i-1)} \Delta \mathbf{U}^{(i)} = {}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F}^{(i-1)}$$

where

$$\mathbf{C} = \alpha \mathbf{M} + \beta \mathbf{K}$$

and $\alpha = \text{ALPHA}$ and $\beta = \text{RESFAC}$ are provided via input.

- As can be seen from the above governing equation, the inertia term is not included, therefore it is beneficial to include the mass term in the damping matrix calculations. A group-based damping can also be used with the Euler-backward method.
- For low speed dynamics with the group-based damping option, the standard Rayleigh damping parameters α and β are used to evaluate the \mathbf{C} matrix, i.e.,

$$\mathbf{C} = \alpha \mathbf{M} + \beta \mathbf{K}$$

- When the low-speed dynamics option is used with ATS, the time step size will influence the results. It is recommended that the time step size be at least $10^5 \beta$. Or, it is recommended that the loads be held constant for a period of time of at least 10β so that the dynamic effects die out.
- Notice that when $\mathbf{C} = \beta \mathbf{K}$, rigid body motions are not damped out. In order to achieve damping of rigid-body motions, $\mathbf{C} = \alpha \mathbf{M} + \beta \mathbf{K}$ must be used.
- Note that the mass matrix and damping matrix can be scaled by time functions when low speed dynamics (implicit dynamics) is used. Time function numbers can be provided in the ANALYSIS command for mass matrix scaling and in the RAYLEIGH-DAMPING command for damping matrix scaling.
- Note that mass effects may not be needed in the low speed dynamic analysis. In this case, set the δ parameter of Eq. (7.2-4) to zero, or, alternatively, set the material densities to zero. That way, only structural damping effects will be present in the otherwise static analysis.

- When there are potential-based fluid elements in the model, the low-speed dynamics option should only be used with the group-based damping option, and there should be no damping assigned to the potential-based fluid element groups.
- ADINA outputs the damping and inertia force norms at every time step (see Section 7.2.5 for more information on these norms). The damping and inertia forces should be compared with the external forces in order to ensure that they are not excessive.
- In low-speed dynamics, a lumped mass matrix is used, unless there are centrifugal, rotational or mass-proportional loads in the model, in which case a consistent mass matrix is used.

7.2.4 Automatic-Time-Stepping (ATS) method

- The automatic-time-stepping (ATS) method controls the time step size in order to obtain a converged solution. If there is no convergence with the user-specified time step, the program automatically subdivides the time step until it reaches convergence. In some cases, the time step size may be increased to accelerate the solution. The ATS method is used by default.
- The user sets the division factor that ADINA uses to subdivide the time step when there is no convergence. Successive subdivisions can be performed, if necessary, provided that the time step size is not smaller than a minimum value. This minimum value is set as the original step size divided by a scaling factor provided by the user.
- Note that the loads at any intermediate time instant created by the ATS method are recalculated based on the current value of the time functions.
- The solution output is only furnished at the user-specified times, except when the solution is abandoned due to too many time step subdivisions without convergence. In this case, the solution output is also given for the last converged time instant.
- There are three options for controlling the time step size once convergence is reached after ATS subdivisions. You can select any

of these options, or request that ADINA does the selection.

1. Use the time increment that gave convergence

In this case, the program continues to use the reduced (subdivided) time step size that gave convergence for the number of consecutive substeps specified by the NSUBSTEP parameter. Once this number of consecutive substeps is reached, or once the end of the user-specified time step is reached, the program might increase the time step size based on the iteration history, but the program ensures that the analysis always proceeds through all user-specified solution times. This option is the default in analyses without contact. Corresponding command input: `AUTOMATIC TIME-STEPPING RESTORE=NO`

2. Return to original time step size

In this case, the program continues the analysis using the original user-provided time increment. This option is the default when contact is present. Corresponding command input: `AUTOMATIC TIME-STEPPING RESTORE=YES`

3. Proceed through user-defined time points

In this case, the program sets the time step size such that the final time is that initially provided by the user. Hence, the analysis always proceeds through all user-specified time steps. Corresponding command input: `AUTOMATIC TIME-STEPPING RESTORE=ORIGINAL`

The program automatically switches to option 3 if option 2 is specified and one of the following features is used: Bathe's composite method, substructuring, FSI, TMC, or mapping.

- The ATS method can also increase the time step beyond the user-specified value if the iteration history shows that such an increase is useful. The increase in time step cannot be larger than the user-specified factor, DTMAX. Due to this increase, the analysis may be completed in fewer time steps than requested. When the ATS subdivision is set to "Return to original step size", the time step acceleration is only possible in a static analysis without low speed dynamics with only one time step block. When the ATS subdivision is set to "Use the time increment that gave convergence", the time step acceleration is possible in all analysis, including analyses with multiple time step blocks.

- Following is an example to illustrate the basic options of ATS subdivisions. Assume we are in load step 15 of a particular problem with initial time $t = 15.0$ and a time step of 1.0. The solution does not converge and the time step is set to 0.5 (assuming a time step division factor of 2.0). If that too does not converge, the time step is set to 0.25. If that converges, the results are not yet saved. Another sub-step is performed for load step 15. The size of this step depends on which of the three options above is selected:
 - ▶ In option 1, the next sub-step will use a time step size of 0.25. Two other sub-steps will be performed within load step 15 both of size 0.25 (assuming they all converge).
 - ▶ In option 2, the next sub-step will use a time step size of 1.0. If this converges load step 16 starts with $t = 16.25$.
 - ▶ In option 3, the sub-step will use a time step size of 0.75. If this converges load step 16 starts with $t = 16.0$.
- Note that while options 1 and 2 may result in outputted solution times that are different from those initially specified by the user, there are certain time values that are not skipped. These are the time values at the end of “time step blocks”. In this case, the time step size is reduced such that the solution time at the end of the block is satisfied. The program determines these time step blocks based on the time step pattern input by the user. The final solution time is always assumed to be an end of a time step block.
- Option 2 is useful for contact because of the highly nonlinear response (sudden changes in stiffness) that occurs when a node comes into contact, or is released from contact, or even moves from one contact segment to another. A small step size may sometimes be needed only in the vicinity of this contact incident. Once contact is established/released the problem is “less nonlinear” and the original time step size can be used.
- Whenever the ATS method is used, the program always checks if a time function point could be skipped for a time step used, and makes an appropriate adjustment to the magnitude of the time step if necessary. Let us consider the time function shown in Fig. 7.2-1.

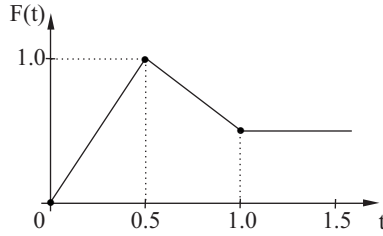


Figure 7.2-1: Time function illustrating ATS program checking

With a time step equal to 0.2, assuming that convergence is obtained, the program will provide solutions for times $t = 0.2$ and $t = 0.4$. If the time step of 0.2 is used again, then the maximum time function value at time $t = 0.5$ will be missed. Therefore, the program uses two substeps of value 0.1 to get to time $t = 0.6$.

- The above option is not available for substructures. In the case of substructures, a constant time stepping is used (ATS is only active when there is no convergence).
- For dynamic nonlinear analysis, the ATS method can also be used to obtain a more accurate solution. The accuracy checking is performed after iteration convergence in each step, using the accuracy criterion

$$\left| {}^{t+\frac{\Delta t}{2}}U_{im} - {}^{t+\frac{\Delta t}{2}}U_{ih} \right|_{\max} \leq \text{DISTOL}$$

in which i represents each translational degree of freedom,

$${}^{t+\frac{\Delta t}{2}}U_{im} = \frac{({}^tU_i + {}^{t+\Delta t}U_i)}{2}$$

$${}^{t+\frac{\Delta t}{2}}U_{ih} = {}^tU_i + \frac{\Delta t}{2}{}^t\dot{U}_i + \frac{\Delta t^2}{16}({}^t\ddot{U}_i + {}^{t+\Delta t}\ddot{U}_i)$$

and ${}^{t+\frac{\Delta t}{2}}U_{ih}$ is the displacement evaluated at time $t + \frac{\Delta t}{2}$.

- Note that all static and dynamic nonlinear analyses, except those only possible with the load-displacement-control (LDC) method, can be carried out with the ATS method.
- The ATS method cannot be used in linearized buckling analysis, mode superposition analysis or response spectrum .

7.2.5 Total Load Application (TLA) method and Stabilized TLA (TLA-S) method

The Total Load Application method is useful for nonlinear static analysis problems where all applied loads do not require the user to explicitly specify the time step sequence. In this case, the user applies the full load value and ADINA automatically applies the load through a ramp time function, and increases or reduces the time step size depending on how well the solution converges. This method cannot be used in dynamic analysis.

The TLA method automatically activates the following features that are suitable for this type of uniform loading static problems:

- The first time step has a size of 1/50 of the total time.
- Maximum number of equilibrium iterations is 30.
- Line search is used.
- Limiting incremental displacements per iteration is set to 5% of the largest model dimension.
- The maximum number of time step subdivisions is set to 64.
- The time step cannot be increased more than 16 times its initial size.

Most of the TLA settings listed above can be modified by the user (via the AUTOMATIC TOTAL-LOAD-APPLICATION command).

The Stabilized TLA method (TLA-S) is identical to the regular TLA method with the addition of various stabilizing features to create a more stable system and aid convergence. The sources of stabilization are low speed dynamics which adds inertia and stiffness proportional damping (see Section 7.2.3), absolute contact damping (see Section 4.4), and stiffness stabilization (see Section 11.14). Indicators are provided in the output file after each

converged solution to show if the forces due to the various stabilization effects are excessive.

The following solution accuracy indicators are printed:

- Internal forces

$$F_{\text{int}} = \sum_1^{\text{neq}} (|RE| + |R_m| + |R_d|)$$

where

RE, R_m, R_d are, respectively, element internal, inertia and damping forces and neq is the total number of equations (excluding contact).

- Internal incremental work

$$\mathbf{I}_{\text{int}} = \mathbf{F}_{\text{int}} \cdot \Delta \mathbf{U}$$

where $\Delta \mathbf{U}$ is a nodal point displacement increment vector

- Damping forces R_d
- Damping incremental work

$$\mathbf{I}_d = \mathbf{R}_d \cdot \mathbf{V}$$

where \mathbf{V} is a nodal point velocity vector

- Inertia forces R_m
- Inertia incremental work

$$\mathbf{I}_m = \mathbf{R}_m \cdot \mathbf{A}$$

where \mathbf{A} is a nodal point acceleration vector

- Contact damping forces \mathbf{R}_{cd}
- Contact damping incremental work

$$\mathbf{I}_{cd} = \mathbf{R}_{cd} \cdot \Delta \mathbf{U}$$

- Stiffness stabilization forces

$$\mathbf{R}_{st} = \tilde{\mathbf{K}} \cdot \Delta \mathbf{U}$$

- Stiffness stabilization incremental work

$$\mathbf{I}_{st} = \mathbf{R}_{st} \cdot \Delta \mathbf{U}$$

- Drilling forces \mathbf{R}_{dr} (applicable to shell elements only)
- Drilling incremental work

$$\mathbf{I}_{dr} = \mathbf{R}_{dr} \cdot \Delta \mathbf{U}$$

The TLA-S method can serve several purposes. If the stabilization indicators are all within reasonable bounds, typically less than 1% of the external force indicator, then the TLA-S solution may be reasonably accurate. However, even when the indicators are large, the TLA-S method provides a useful approximate solution that can frequently be used to detect modeling errors such as incorrect contact definition, applied load, or boundary conditions.

The following features cannot be used with the TLA and TLA-S methods:

- All materials with creep effects (applies to elastic and plastic creep materials)
- All materials with viscosity effects (visco-elastic material and Anand solder material)
- User-supplied materials
- Rigid target contact

7.2.6 Load-Displacement-Control (LDC) method

- The load-displacement-control (LDC) method (arc length method) can be used to solve for the nonlinear equilibrium path of a model until its collapse. If desired, the post-collapse response of the model can also be calculated. The main feature of the method is that the level of the externally applied loads is adjusted automatically by the program.

The LDC method can only be used in nonlinear static analysis in which there are no strain-rate, pipe internal pressure or creep effects. The LDC method can be used in contact problems.

- The formulations used in the LDC method in ADINA are described in ref. KJB Section 8.4.3 and in the following reference:

ref. Bathe, K.J. and Dvorkin, E.N., "On the Automatic Solution of Nonlinear Finite Element Equations," *J. Computers and Structures*, Vol. 17, No. 5-6, pp. 871-879, 1983.

- The LDC method is activated via AUTOMATIC=LDC in the MASTER command. An enforced displacement on a user-specified degree of freedom is used to evaluate the initial load vector, and analysis continues until a specified displacement is reached at a certain node, or a critical point on the equilibrium path is reached. The LDC method may also be used as a restart analysis following a static or dynamic analysis. Variants of the LDC method are commonly referred to as arc-length methods.

- The equations employed in the equilibrium iterations are

$$\begin{aligned}
 {}^{t+\Delta t}\mathbf{K}^{(i-1)} \Delta \mathbf{U}^{(i)} &= \left({}^{t+\Delta t}\lambda^{(i-1)} + \Delta \lambda^{(i)} \right) \mathbf{R} + \mathbf{R}_p - {}^{t+\Delta t}\mathbf{F}^{(i-1)} \\
 {}^{t+\Delta t}\mathbf{U}^{(i)} &= {}^{t+\Delta t}\mathbf{U}^{(i-1)} + \Delta \mathbf{U}^{(i)} \\
 f\left(\Delta \lambda^{(i)}, \Delta \mathbf{U}^{(i)}\right) &= 0
 \end{aligned} \tag{7.2-5}$$

where

$${}^{t+\Delta t}\mathbf{K}^{(i-1)} = \text{tangent stiffness matrix at time } t+\Delta t, \text{ end of}$$

- iteration ($i-1$)
- \mathbf{R} = constant reference load vector
- \mathbf{R}_p = load vector from previous solution run
- ${}^{t+\Delta t}\lambda^{(i-1)}$ = load scaling factor (used on \mathbf{R}) at the end of iteration ($i-1$) at time $t+\Delta t$
- $\Delta\lambda^{(i)}$ = increment in the load scaling factor in iteration (i)

The quantities ${}^{t+\Delta t}\mathbf{F}^{(i-1)}$ and $\Delta\mathbf{U}^{(i)}$ are as defined for Eq. (7.2-1).

Note that in Eq. (7.2-5), the equation $f=0$ is used to constrain the length of the load step. The constant spherical arc length constraint method is usually used and the constant increment of external work method is used if the arc length method has difficulties to converge.

The reference load vector \mathbf{R} is evaluated from all the mechanical loads (except user-supplied and prescribed displacement loadings).

- In a restart LDC analysis, the load vector from the previous solution run \mathbf{R}_p may be held constant. This is achieved by requesting, in the first run, that the load vector be transferred to the restart file. This feature is useful, for example, to model varying loads in the presence of a constant gravity load. In the first run, enter only the gravity loading and request the load transfer. Then, in the restart run, add the varying loads using the LDC method.
- To start the LDC method, the load multiplier for the first step ($\Delta\lambda$), (used to obtain the corresponding load vector $\Delta\lambda \mathbf{R}$) is calculated using a user-specified prescribed displacement. This prescribed displacement (see Fig. 7.2-1) acts in the specified direction at the specified node. The direction of the displacement is given by its sign.

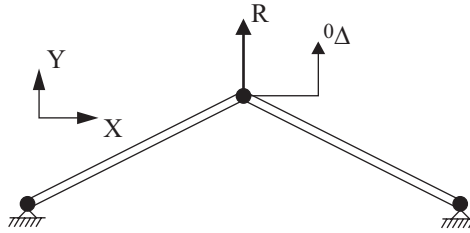
Note that the input for the initial prescribed displacement (in particular whether it is positive or negative) is critical in establishing successive equilibrium positions using the LDC method.

As an example, two entirely different solution paths will be obtained for the same model shown in Fig. 7.2-1 if initial

displacements of different signs are prescribed for the first solution step.

- After the first step, the program automatically traces the nonlinear response by scaling the external load vector \mathbf{R} proportionally, subject to the constraint of Eq. (7.2-5), so that at any discrete time t in iteration (i) , the external load vector is ${}^t\lambda^{(i)}\mathbf{R}$.

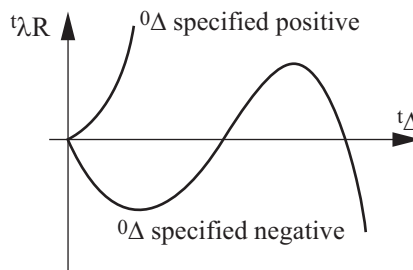
The scaling of the reference load vector using ${}^t\lambda$ is analogous to the scaling of the applied loads \mathbf{R} using a user predefined time function when the LDC method is not used (see Chapter 5). In the case of the LDC method, the scaling function is determined internally by the program instead of being user-specified.



Reference load = \mathbf{R} , actual load at time $t = {}^t\lambda\mathbf{R}$

Prescribed displacement for first step = ${}^0\Delta$, displacement at time $t = {}^t\Delta$

a) Model considered



b) Equilibrium paths

Figure 7.2-2: Example of the dependence of solution path on the displacement prescribed in the first step for the LDC method

- The converged displacement must satisfy the following relation:

$$\|\mathbf{U}\|_2 \leq 100\alpha \|\Delta^t \mathbf{U}\|_2$$

where $\mathbf{U} = {}^{t+\Delta t}\mathbf{U} - {}^t\mathbf{U}$ is the incremental displacement vector for the current step, α is a displacement convergence input factor and $\Delta^t \mathbf{U}$ is the displacement vector obtained in the first step. If the above inequality is not satisfied, an internal restart of the iteration for the current step is performed by the program.

- The LDC solution terminates normally when any one of the following conditions is satisfied:
 - ▶ The maximum displacement that you specify is reached.
 - ▶ A critical point on the equilibrium path has been passed and you request solution termination.
 - ▶ The number of converged solution steps is reached.
 - ▶ The incremental solution has been attempted a number of times as specified by input (default is ten) from a calculated equilibrium configuration using different strategies but each time the solution has failed to converge within the number of allowed iterations.

7.2.7 Convergence criteria for equilibrium iterations

- The following convergence criteria can be specified in ADINA:
 - ▶ energy only,
 - ▶ energy and force/moment,
 - ▶ energy and translation/rotation,
 - ▶ force/moment only, and
 - ▶ translation/rotation only.

- If contact surfaces are defined in an analysis, then the contact force convergence criterion is always used in addition to the above criteria (see Chapter 4).
- The values of all convergence norms, whether used or not, are provided the .out file. For more details on the .out file format see Section 7.2.9.
- When ADINA runs in interactive mode, graphical output of all convergence norms is also provided.

LDC method not used

- If the LDC method is not used, the convergence in equilibrium iterations is reached when the following inequalities are satisfied:

Energy convergence criterion: For all degrees of freedom

$$\frac{\Delta \mathbf{U}^{(i)T} \left[{}^{t+\Delta t} \mathbf{R} - {}^{t+\Delta t} \mathbf{F}^{(i-1)} \right]}{\Delta \mathbf{U}^{(1)T} \left[{}^{t+\Delta t} \mathbf{R} - {}^t \mathbf{F} \right]} \leq \text{ETOL} \quad (7.2-6)$$

where ETOL is a user-specified energy convergence tolerance.

Force and moment convergence criteria: For the translational degrees of freedom

$$\frac{\left\| {}^{t+\Delta t} \mathbf{R} - {}^{t+\Delta t} \mathbf{F}^{(i-1)} \right\|_2}{\text{RNORM}} \leq \text{RTOL}$$

For the rotational degrees of freedom

$$\frac{\left\| {}^{t+\Delta t} \mathbf{R} - {}^{t+\Delta t} \mathbf{F}^{(i-1)} \right\|_2}{\text{RMNORM}} \leq \text{RTOL}$$

where RTOL is a user-specified force convergence tolerance, RNORM and RMNORM are user-specified reference force and moment norms. If left undefined the program automatically determines RNORM and RMNORM during execution.

Translation/rotation convergence criteria: For the translational degrees of freedom

$$\frac{\|\Delta \mathbf{U}^{(i)}\|_2}{\text{DNORM}} \leq \text{DTOL}$$

For the rotational degrees of freedom

$$\frac{\|\Delta \mathbf{U}^{(i)}\|_2}{\text{DMNORM}} \leq \text{DTOL}$$

where DTOL is a user-specified force convergence tolerance, DNORM and DMNORM are user-specified reference displacement and rotation norms. If left undefined the program automatically determines DNORM and DMNORM during execution.

Note that in each of these convergence criteria the residual norm is measured against a user-specified maximum residual value; for example, the force criterion could be interpreted as

$$(\text{norm of out-of-balance loads}) \leq \text{RTOL} \times \text{RNORM}$$

where $\text{RTOL} \times \text{RNORM}$ is equal to the user-specified maximum allowed out-of-balance load.

Note also that these convergence criteria are used in each subdivision of time or load step when the ATS method of automatic step incrementation is used.

- If contact surface groups are present in the analysis the following additional criterion is always used in measuring convergence

$$\frac{\max\left(\|\mathbf{R}_c^{(i-1)} - \mathbf{R}_c^{(i-2)}\|_2, \|\Delta \boldsymbol{\lambda}^{(i)} - \Delta \boldsymbol{\lambda}^{(i-1)}\|_2\right)}{\max\left(\|\mathbf{R}_c^{(i-2)}\|_2, \text{RCONSM}\right)} \leq \text{RCTOL}$$

where $\mathbf{R}_c^{(i-1)}$ is the contact force vector at the end of iteration

$(i-1)$, $\Delta\lambda^{(i)}$ is the incremental Lagrange multiplier vector at the end of iteration i , RCONSM is a reference contact force level used to prevent possible division by zero and RCTOL is a user-specified contact force convergence tolerance.

Non-convergence: Convergence might not occur when the maximum number of iterations is reached or when the solution is diverging.

- If the specified convergence criteria are not satisfied within the allowed number of iterations, but the solution is not diverging, the following can be attempted:

- ▶ Check the model according to the guidelines in Section 7.2.6.
- ▶ Use a smaller time step.
- ▶ Increase the number of allowable iterations.
- ▶ Change the ATS parameters.
- ▶ Change convergence tolerances. In most cases, looser tolerances help. However, in some problems, tighter tolerances help by not allowing approximate solutions that could potentially prevent convergence in future time steps.
- ▶ Change line search. Some problems, such as those involving plasticity, perform better with line search.
- ▶ Change contact settings. The optimal contact settings and features depend on the model. See Chapter 4 for more details.

- Divergence of solution terminates the iteration process before the maximum number of iterations is reached. It is sometimes detected when the energy convergence ratio in Eq. (7.2-6) becomes unacceptably large, or when the excessive displacements lead to distorted elements and negative Jacobians. The program will repeat the iteration number at which divergence occurs. In this case, the following can be attempted:

- ▶ Check the model according to the guidelines in Section 7.2.6.
- ▶ Use a smaller time step.
- ▶ Make sure there are sufficient constraints to remove rigid body modes from all components in the model. Presence of rigid body modes usually results in a large ratio of maximum to minimum pivot during factorization (with sparse solver).

LDC method used

- Convergence in the equilibrium iterations is reached when the following inequalities are satisfied:

Energy convergence criterion: For all degrees of freedom

$$\frac{\Delta \mathbf{U}^{(i)T} \left[{}^{t+\Delta t} \lambda^{(i-1)} \mathbf{R} - {}^{t+\Delta t} \mathbf{F}^{(i-1)} \right]}{\Delta \mathbf{U}^{(1)T} \left[\Delta \lambda^{(1)} \mathbf{R} \right]} \leq \text{ETOL}$$

where ETOL is a user-specified energy convergence tolerance.

Force and moment convergence criteria: For the translational degrees of freedom

$$\frac{\left\| {}^{t+\Delta t} \lambda^{(i-1)} \mathbf{R} - {}^{t+\Delta t} \mathbf{F}^{(i-1)} \right\|_2}{\text{RNORM}} \leq \text{RTOL}$$

For the rotational degrees of freedom

$$\frac{\left\| {}^{t+\Delta t} \lambda^{(i-1)} \mathbf{R} - {}^{t+\Delta t} \mathbf{F}^{(i-1)} \right\|_2}{\text{RMNORM}} \leq \text{RTOL}$$

where RTOL is a user-specified force convergence tolerance, RNORM and RMNORM are user-specified reference force and moment norms. If left undefined the program automatically determines RNORM and RMNORM during execution.

The translation/rotation convergence criteria, and the contact convergence criterion, are the same as when the LDC method is not used, see above.

Nonconvergence: If convergence has not been reached from an established equilibrium configuration after the maximum restart attempts, the program saves the required restart information and program execution is terminated.

The solution can be continued by performing a restart run. Note that in this case the LDC method must be used in the restart run. A different value for the initial displacement can be prescribed at a different nodal point in the first step of the restart run. The prescribed initial displacement then corresponds to a displacement increment from the last converged equilibrium position, that is, at the time of solution start for the restart analysis.

7.2.8 Selection of incremental solution method

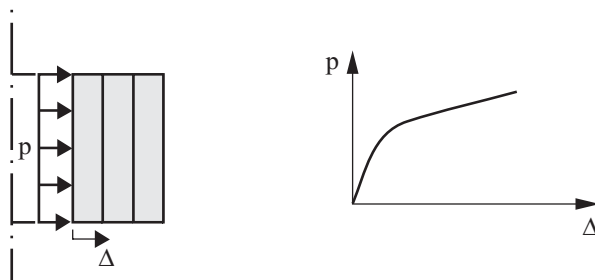
- This section gives recommendations on which incremental solution method to use for a given analysis.
- Every nonlinear analysis should be preceded by a linear analysis, if only to check that the model has been set up correctly. The linear analysis results will tell whether proper boundary conditions are imposed, all degrees of freedom without stiffness have been deleted, and so on (see Section 7.1), and the finite element mesh is adequate.

In practice, it may be best to set up the complete nonlinear model, but only perform a one load step analysis without equilibrium iterations. The displacement results obtained from this solution are linear analysis results and can frequently be used to establish a reasonable load incrementation for the nonlinear solution.

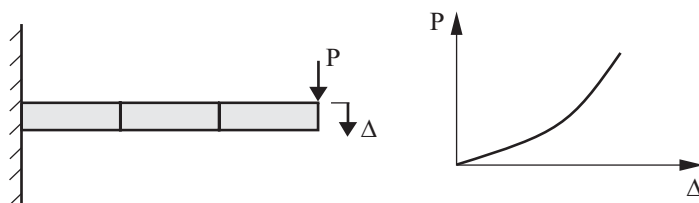
- If the use of a sufficient number of load steps and equilibrium iterations with tight convergence tolerances at each load step is considered to yield an accurate solution of the model, then the basic aim is to obtain a response prediction close to this accurate one at as small a solution cost as possible.

- If the cost of solution without equilibrium iterations seems reasonable and reasonable nonlinearities per step were observed, it is usually best to perform the next analysis with equilibrium iterations using the default iteration technique, the full Newton method, and default tolerances of the program.
- It is helpful to know if the model softens or hardens under increasing load. Structures can soften due to the spread of plasticity, and they can soften or harden due to geometric nonlinear effects. Contact usually leads to hardening. Fig. 7.2-3 shows some examples.
- Displacement controlled loading generally converges faster than force controlled loading. For example, in the model shown in Fig. 7.2-3(b), applying an increasing tip displacement to the beam will converge faster than an applied load P , and both will follow the same load displacement curve. For the model in Fig. 7.2-3(c) force control would fail past the local maximum on the load-displacement curve. Displacement controlled loading (apply an increasing Δ) would work in this case. Note that this case is also suitable for the LDC method.
- The use of the full Newton iteration method and a reasonable load incrementation is frequently sufficient to ensure that an accurate solution of the response of the model will be obtained.
- The ATS method can be used with the modified Newton iteration, the BFGS method and the full Newton method. If a reasonable number of time (or load) steps is used for the nonlinear response, then the use of the ATS method will result in almost the same "iteration path" as when not using the method. Namely, no step subdividing will be performed if convergence is always directly reached at the user-specified load levels. Hence, in general, if you want to use the most convenient method, it is recommended to use a reasonable number of load steps with the default iteration scheme (full Newton method), with default tolerances and the ATS method. However, a less expensive solution can frequently be reached by use of the other available iteration methods.
- The LDC method is very useful if collapse of the structure

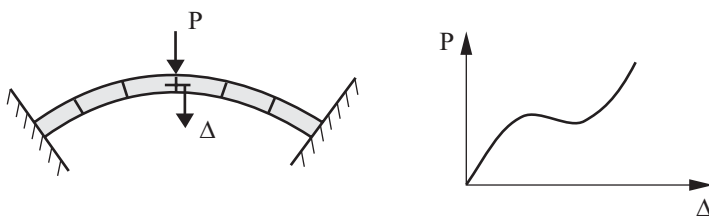
occurs during the (static) solution. However, the use of this algorithm can be very costly. The LDC method is recommended if you do not want to specify the load increments for the solution period and computational costs are not of primary concern. The LDC method allows also the calculation of the post-collapse response. Note, however, that the displacement solution at a specific load level cannot be obtained using the LDC method because the load increments are calculated by the program.



(a) Softening problem. Materially-nonlinear-only analysis, elasto-plastic analysis of a cylinder



(b) Stiffening problem. Large displacement nonlinear elastic analysis of a cantilever



(c) Softening/stiffening problem. Large displacement analysis of a thin arch

Figure 7.2-3: Different types of nonlinear analyses

- Note that usually it is quite adequate to employ the energy convergence tolerance only. The need to use the force/moment criteria and/or translation/rotation criteria arises when the energy convergence is not tight (small) enough. In addition, there exist special loading conditions under which the denominator of the inequality (7.2-6) in Section 7.2.7 becomes small and hence the inequality is difficult to satisfy.

7.2.9 Example

```

STEP NUMBER = 4 ( TIME STEP = 0.20000E-03 SOLUTION TIME = 0.80000E-03 ) DOF = 2-TRANSLATION
DIAG ELEMENT (WITH MAX ABS VALUE) OF THE FACTORIZED MATRIX = 0.5547E+10 NOSE = 740 DOF = X-ROTATION
DIAG ELEMENT (WITH MIN ABS VALUE) OF THE FACTORIZED MATRIX = 0.5841E+02 NOSE = 161 DOF = X-ROTATION

METHOD STEP-NUMBER SUBINCREMENT TIME STEP SOLUTION TIME INITIAL ENERGY
*ATS* 4 1 0.20000E-03 0.80000E-03 0.223798E+04

INTERMEDIATE PRINTOUT DURING EQUILIBRIUM ITERATIONS

CONVERGENCE RATIOS CONVERGENCE RATIOS OUT-OF-BALANCE LOAD
FOR OUT-OF-BALANCE FOR INCREMENTAL VECTOR CALCULATION
ENERGY FORCE DISP. CFORCE BETA RATIO
MOMENT ROTN. (ITERNS)
COMPARE WITH COMPARE WITH
ETOL RTOL DTOL RTOL
1.00E-06 1.00E-02 (NOT USED) 1.00E-03

ITE= 0 2.24E+03 2.32E+06 3.27E-03 4.79E-03 7.45E-01 4.74E+03 1.00E+00 2.32E+05 6.92E-02 1.15E+00
740-Z 740-Z 159-Z 239-Y 4.11E+03
-2.32E+06 -6.22E-04 -1.14E-03 1.64E-01 3.27E-04 8.63E-01

ITE= 1 3.95E+00 1.96E+04 9.93E+00 6.16E-04 2.03E-01 4.74E+03 1.77E-03 1.96E+03 8.90E-03 9.01E-01 9.99E-01 -5.95E-08
239-X 239-X 39-Y 239-Y 5.26E+03 9.93E-01 2.35E-01 ( 3 )
4.14E+03 -2.37E+00 1.52E-04 6.71E-02

ITE= 2 2.67E-01 5.89E+03 4.45E+00 1.29E-04 5.95E-02 4.54E+02 1.19E-04 5.89E+02 1.86E-03 9.06E-02 4.94E-01 2.85E-06
-1.35E+03 -1.16E+00 -1.67E-05 -1.95E-02 4.45E-01 6.92E-02 ( 5 )

ITE= 3 4.28E-03 1.00E+03 5.20E-01 2.45E-05 7.30E-03 4.46E+02 1.91E-06 1.00E+02 3.53E-04 9.07E-02 9.81E-01 -3.45E-05
39-Z 472-Y 39-X 79-Y 4.61E+03 5.20E-02 8.45E-03 ( 3 )
-2.80E+02 1.72E-01 1.72E-01 3.53E-06 -1.86E-03

ITE= 4 2.48E-06 1.76E+01 1.35E-02 3.47E-07 1.71E-04 3.12E+01 1.11E-09 1.77E+00 5.01E-06 6.50E-03 1.00E+00 8.78E-03
39-Z 239-Y 159-Z 119-Y 4.58E+03 1.35E-03 1.98E-04 ( 1 )
-5.76E+00 -5.12E-03 -1.16E-07 -5.42E-05

ITE= 5 8.61E-10 1.53E-04 1.14E-08 6.76E-06 1.99E-01 3.85E-13 1.51E-02 1.65E-07 4.35E-05 1.00E+00 1.54E-03
79-Z 119-Y 159-Z 119-Y 4.58E+03 1.53E-05 7.83E-06 ( 1 )
5.41E-02 7.55E-05 3.59E-09 2.66E-06

ITE= 6 1.87E-12 5.22E-03 9.02E-06 4.84E-10 2.62E-07 2.64E-02 8.35E-16 5.22E-04 6.99E-09 5.76E-06 1.00E+00 -2.39E-02
119-Z 119-Y 159-Z 119-Y 4.58E+03 9.02E-07 3.03E-07 ( 1 )
2.05E-03 -4.21E-06 -1.80E-10 -1.30E-07

```

Figure 7.2-4: Example of iteration history printout

```

I T E R A T I O N   T I M E   L O G

SOLUTION TIME (SECONDS) . . . . . = 5.13

PERCENT OF TIME SPENT IN LINE SEARCHING . . . . . = 54.00
PERCENT OF TIME SPENT IN LOAD VECTOR/STIFFNESS MATRIX CALCULATION . = 39.57
PERCENT OF TIME SPENT IN SOLUTION OF EQUATIONS . . . . . = 6.43

6 EQUILIBRIUM ITERATIONS PERFORMED IN THIS TIME STEP TO REESTABLISH EQUILIBRIUM
STIFFNESS REFORMED FOR EVERY ITERATION OF THIS STEP
NUMBER OF SUBINCREMENTS IN THIS TIME STEP = 1

```

Figure 7.2-4: (continued)

This section presents a worked example that illustrates the nonlinear iteration and convergence concepts previously discussed. Figure 7.2-4 shows the iteration history for a load step. The standard Newton method with line searches is used with the energy and force convergence criterion (ETOL= 1×10^{-6} , RTOL=0.01, RNORM=10.0, RMNORM=10.0). For contact RCTOL= 1×10^{-3} and the reference contact force RCONSM=0.01. For line search STOL=0.01.

Row ITE=0: This row shows the result of the initial iteration called iteration 0. For this iteration, the program performs the following steps:

Compute ${}^{t+\Delta t}\mathbf{U}^{(0)} = {}^t\mathbf{U}$.

Compute ${}^{t+\Delta t}\mathbf{F}^{(0)}$ and ${}^{t+\Delta t}\mathbf{K}^{(0)}$ using ${}^{t+\Delta t}\mathbf{U}^{(0)}$.

Compute the out-of-balance force vector ${}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F}^{(0)}$. Only considering translational degrees of freedom, the norm of the out-of-balance force vector is $\|{}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F}^{(0)}\|_2 = 2.32 \times 10^6$ and the largest magnitude in the out-of-balance force vector is -2.32×10^6 at the Z translational degree of freedom of node 740. Only considering rotational degrees of freedom, the norm of the out-of-balance force vector is $\|{}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F}^{(0)}\|_2 = 3.27 \times 10^{-3}$ and the largest magnitude in the out-of-balance force vector is -6.22×10^{-4} at the Y rotational degree of freedom of node 319.

Compute $\Delta\mathbf{U}^{(0)}$ using ${}^{t+\Delta t}\mathbf{K}^{(0)}\Delta\mathbf{U}^{(0)} = {}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F}^{(0)}$. Only considering translational degrees of freedom, the norm of the

incremental displacement vector is $\|\Delta \mathbf{U}^{(0)}\|_2 = 4.79 \times 10^{-3}$ and the largest magnitude in the incremental displacement vector is -1.14×10^{-3} at the Z displacement of node 159. Only considering rotational degrees of freedom, the norm of the incremental displacement vector is $\|\Delta \mathbf{U}^{(0)}\|_2 = 7.45 \times 10^{-1}$ and the largest magnitude in the incremental displacement vector is 1.64×10^{-1} at the Y rotation of node 239.

Computes the “out-of-balance energy”

$$\Delta \mathbf{U}^{(0)T} \left({}^{t+\Delta t} \mathbf{R} - {}^{t+\Delta t} \mathbf{F}^{(0)} \right) = 2.24 \times 10^3.$$

Compute the norm of the change in contact forces

CFORCE = 4.74×10^3 , and the norm of the contact forces
CFNORM = 4.11×10^3 .

Compute the energy convergence criterion

$$\frac{\Delta \mathbf{U}^{(0)T} \left[{}^{t+\Delta t} \mathbf{R} - {}^{t+\Delta t} \mathbf{F}^{(0)} \right]}{\Delta \mathbf{U}^{(0)T} \left[{}^{t+\Delta t} \mathbf{R} - {}^t \mathbf{F} \right]} = 1.00$$

Compute the force and moment convergence criteria

$$\frac{\| {}^{t+\Delta t} \mathbf{R} - {}^{t+\Delta t} \mathbf{F}^{(0)} \|_2}{\text{RNORM}} = 2.32 \times 10^5 \text{ and}$$

$$\frac{\| {}^{t+\Delta t} \mathbf{R} - {}^{t+\Delta t} \mathbf{F}^{(0)} \|_2}{\text{RMNORM}} = 3.27 \times 10^{-4}.$$

Compute the contact convergence criterion

$$\frac{\text{CFORCE}}{\max(\text{CFNORM}, \text{RCONSM})} = 1.15 \times 10^0$$

The energy convergence criterion is greater than ETOL, the force convergence criterion is greater than RTOL and the contact convergence criterion is greater than RCTOL. Therefore, convergence is not satisfied.

Note that the displacement and rotation norms are also substituted into the displacement convergence criterion, which results in

$$\frac{\|\Delta \mathbf{U}^{(0)}\|_2}{\text{DNORM}} = 6.92 \times 10^{-2} \text{ and}$$

$$\frac{\|\Delta \mathbf{U}^{(0)}\|_2}{\text{DMNORM}} = 8.63 \times 10^{-1}.$$

Since DNORM and DMNORM are not provided by the user, they are automatically estimated by the program. The above displacement convergence values however are not used in determining convergence.

Row ITE=1: This row shows the results of the first equilibrium iteration. In this iteration, the program performs the following steps:

Compute ${}^{t+\Delta t}\mathbf{U}^{(1)} = {}^{t+\Delta t}\mathbf{U}^{(0)} + \Delta \mathbf{U}^{(0)}$, ${}^{t+\Delta t}\mathbf{F}^{(1)}$ and the line search ratio $\frac{\Delta \mathbf{U}^{(0)T} [{}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F}^{(1)}]}{\Delta \mathbf{U}^{(0)T} [{}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F}^{(0)}]}$. This ratio turns out to

be greater than STOL=0.01, so line search is performed for 3 steps and ends up with a line search energy ratio of 5.95×10^{-8} which is less than STOL, corresponding to $\beta^{(1)} = 0.999$.

Compute ${}^{t+\Delta t}\mathbf{U}^{(1)} = {}^{t+\Delta t}\mathbf{U}^{(0)} + \beta^{(1)}\Delta \mathbf{U}^{(0)}$.

Compute ${}^{t+\Delta t}\mathbf{F}^{(1)}$ using ${}^{t+\Delta t}\mathbf{U}^{(1)}$, and compute ${}^{t+\Delta t}\mathbf{K}^{(1)}$.

Note that the stiffness matrix is updated since the standard Newton method is used.

Compute the out-of-balance force vector ${}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F}^{(1)}$. Only considering translational degrees of freedom, the norm of the out-of-balance force vector is $\|{}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F}^{(1)}\|_2 = 1.96 \times 10^4$ and the largest magnitude in the out-of-balance force vector is

4.14×10^3 at node 239 (X translation). Only considering rotational degrees of freedom, the norm of the out of balance force vector is $\left\| {}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F}^{(1)} \right\|_2 = 9.93 \times 10^0$ and the largest magnitude in the out-of-balance force vector is -2.37×10^0 at node 39 (Y rotation).

Compute $\Delta \mathbf{U}^{(1)}$ using ${}^{t+\Delta t}\mathbf{K}^{(1)}\Delta \mathbf{U}^{(1)} = {}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F}^{(1)}$. Only considering translational degrees of freedom, the norm of the incremental displacement vector is $\left\| \Delta \mathbf{U}^{(1)} \right\|_2 = 6.16 \times 10^{-4}$ and the largest magnitude in the incremental displacement vector is 1.52×10^{-4} at node 239 (X translation). Only considering rotational degrees of freedom, the norm of the incremental displacement vector is $\left\| \Delta \mathbf{U}^{(1)} \right\|_2 = 2.03 \times 10^{-1}$ and the largest magnitude in the incremental displacement vector is 6.71×10^{-2} at node 279 (Y rotation).

Compute $\text{CFORCE} = 4.74 \times 10^3$ and $\text{CFNORM} = 5.26 \times 10^3$.

Compute the “out-of-balance energy”

$$\Delta \mathbf{U}^{(1)T} \left({}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F}^{(1)} \right) = 3.95 \times 10^0.$$

Compute the energy convergence criterion

$$\frac{\Delta \mathbf{U}^{(1)T} \left[{}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F}^{(1)} \right]}{\Delta \mathbf{U}^{(1)T} \left[{}^{t+\Delta t}\mathbf{R} - {}^t\mathbf{F} \right]} = 1.77 \times 10^{-3}.$$

Compute the force and moment convergence criteria

$$\frac{\left\| {}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F}^{(1)} \right\|_2}{\text{RNORM}} = 1.96 \times 10^3 \text{ and}$$

$$\frac{\left\| {}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F}^{(1)} \right\|_2}{\text{RMNORM}} = 9.93 \times 10^{-1}.$$

Compute the contact convergence criterion

$$\frac{\text{CFORCE}}{\max(\text{CFNORM}, \text{RCONSM})} = 9.01 \times 10^{-1}$$

The energy convergence criterion is greater than

ETOL, the force convergence criterion is greater than RTOL, the moment convergence criterion is greater than RTOL, and the contact convergence criterion is greater than RCTOL. Therefore, convergence is not satisfied.

The displacement convergence criterion is also evaluated for informational purposes.

Row ITE=2: This row shows the results from the second equilibrium iteration. This row is interpreted exactly as is row ITE=1. The line search factor in this case is 4.84×10^{-1} obtained in 5 line search iterations.

Again, none of the convergence criteria is satisfied. However, they are all getting smaller.

Row ITE=3: This row shows the results from the third equilibrium iteration. This row is interpreted exactly as is row ITE=2.

Again, none of the convergence criteria is satisfied.

Row ITE=4: This row shows the results from the fourth equilibrium iteration. In this case, the previous increment of displacement from the solver $\Delta \mathbf{U}^{(3)}$ satisfies the line search energy tolerance STOL so no line search is performed.

$${}^{t+\Delta t} \Delta \mathbf{U}^{(4)} = {}^{t+\Delta t} \Delta \mathbf{U}^{(3)} + \Delta \mathbf{U}^{(3)}$$

At the end of the that equilibrium iteration, the energy convergence criterion is

$$\frac{\Delta \mathbf{U}^{(4)T} \left[{}^{t+\Delta t} \mathbf{R} - {}^{t+\Delta t} \mathbf{F}^{(4)} \right]}{\Delta \mathbf{U}^{(4)T} \left[{}^{t+\Delta t} \mathbf{R} - {}^t \mathbf{F} \right]} = 1.11 \times 10^{-9} \text{ which is less than}$$

ETOL. The force convergence value is 1.77×10^0 and the moment convergence value is 1.35×10^{-3} . The contact convergence value is 6.80×10^{-3} .

Two of these four criteria (force and contact) are not satisfied, so convergence is not satisfied.

Row ITE=5: The row shows the results for the fifth equilibrium iteration. Contact convergence criterion is now also satisfied leaving only force convergence unsatisfied. The solution continues.

Row ITE=6: The row shows the results from the sixth equilibrium iteration. In this case, all convergence criteria are satisfied and convergence is reached.

7.3 Linear dynamic analysis

- The following procedures are available in ADINA for solution of the finite element equations in a linear dynamic analysis:

Step-by-step direct integration

- ▶ Explicit time integration using the central difference method or the Noh-Bathe method (described in Chapter 8)
- ▶ Implicit time integration using the Newmark method or the Bathe method

Time history mode superposition analysis

- ▶ Implicit time integration using the Newmark method (trapezoidal rule)

In this case, the eigenproblem for the vibration of the structural system is solved for the natural frequencies and mode shapes (see Chapter 6 for details).

- The notation given below is used in the following sections in the descriptions of the equilibrium equations:

$$\begin{aligned}\mathbf{M} &= \text{constant mass matrix} \\ \mathbf{C} &= \text{constant damping matrix} \\ \mathbf{K} &= \text{constant stiffness matrix} \\ {}^t\mathbf{R}, {}^{t+\Delta t}\mathbf{R} &= \text{external load vector applied at time } t, t+\Delta t \\ {}^t\mathbf{F} &= \text{nodal point force vector equivalent to the element stresses at time } t \\ {}^t\dot{\mathbf{U}}, {}^{t+\Delta t}\dot{\mathbf{U}} &= \text{vectors of nodal point velocities at time } t, t+\Delta t\end{aligned}$$

- ${}^t\ddot{\mathbf{U}}, {}^{t+\Delta t}\ddot{\mathbf{U}}$ = vectors of nodal point accelerations at time t , $t+\Delta t$
 \mathbf{U} = vector of nodal point displacement increments from time t to time $t+\Delta t$, i.e., $\mathbf{U} = {}^{t+\Delta t}\mathbf{U} - {}^t\mathbf{U}$.

7.3.1 Step-by-step implicit time integration

- The governing equilibrium equation at time $t+\Delta t$ is given by

$$\mathbf{M} {}^{t+\Delta t}\ddot{\mathbf{U}} + \mathbf{C} {}^{t+\Delta t}\dot{\mathbf{U}} + \mathbf{K} {}^{t+\Delta t}\mathbf{U} = {}^{t+\Delta t}\mathbf{R} \quad (7.3-1)$$

The procedures used in the time integration of the governing equations for dynamic analysis are summarized in ref. KJB, Chapter 9.

The Newmark method and the Bathe method are summarized in ref. KJB., Sections 9.2.3 and 9.2.4.

- The time integration of the governing equations can use the Newmark method or the Bathe method, with the Bathe method being the default. The Bathe method is explained in the following paper.

ref. K.J. Bathe, "Conserving Energy and Momentum in Nonlinear Dynamics: A Simple Implicit Time Integration Scheme" *J. Computers and Structures*, Vol. 85, Issue 7-8, pp. 437-445. (2007)

- In the standard Bathe method, the time increment Δt is divided into two sub-steps for each time integration step Δt . In the first sub-step, the displacements, velocities, and accelerations are solved for at a time $t + \gamma\Delta t$, where $\gamma \in (0,1)$ using the Newmark trapezoidal rule. The γ parameter is always set to $2 - \sqrt{2} \approx 0.5858$ to keep the same effective stiffness matrix in the two substeps to avoid recalculating and refactorizing that matrix. In the second sub-step a 3-point Euler backward method is used to solve for the displacements, velocities and accelerations at time $t + \Delta t$ using the results at time t and $t + \gamma\Delta t$.

Note that, for a given step size, the Bathe scheme is about twice as expensive as the Newmark method, or trapezoidal rule, due to the

extra solution step at time $t + \gamma\Delta t$, but the cost is compensated for by using less time steps

- The following assumptions are used in the Newmark method:

$${}^{t+\Delta t}\dot{\mathbf{U}} = {}^t\dot{\mathbf{U}} + \left[(1-\delta) {}^t\ddot{\mathbf{U}} + \delta {}^{t+\Delta t}\ddot{\mathbf{U}} \right] \Delta t \quad (7.3-2)$$

$${}^{t+\Delta t}\mathbf{U} = {}^t\mathbf{U} + {}^t\dot{\mathbf{U}} \Delta t + \left[\left(\frac{1}{2} - \alpha \right) {}^t\ddot{\mathbf{U}} + \alpha {}^{t+\Delta t}\ddot{\mathbf{U}} \right] \Delta t^2 \quad (7.3-3)$$

where α and δ are the Newmark time integration parameters.

This transforms Eq. (7.3-1) to

$$\hat{\mathbf{K}} {}^{t+\Delta t}\mathbf{U} = {}^{t+\Delta t}\hat{\mathbf{R}} \quad (7.3-4)$$

where

$$\hat{\mathbf{K}} = \mathbf{K} + a_0\mathbf{M} + a_1\mathbf{C} \quad (7.3-5)$$

$$\hat{\mathbf{R}} = {}^{t+\Delta t}\mathbf{R} + \mathbf{M} \left(a_0 {}^t\mathbf{U} + a_2 {}^t\dot{\mathbf{U}} + a_3 {}^t\ddot{\mathbf{U}} \right) + \mathbf{C} \left(a_1 {}^t\mathbf{U} + a_4 {}^t\dot{\mathbf{U}} + a_5 {}^t\ddot{\mathbf{U}} \right) \quad (7.3-6)$$

and where a_0, a_1, \dots, a_5 are integration constants for the Newmark method (see Ref. KJB, Section 9.2.4).

- The following assumptions are used in the Bathe method:

For the first sub-step we use:

$${}^{t+\Delta t/2}\dot{\mathbf{U}} = {}^t\dot{\mathbf{U}} + \left[\frac{\Delta t}{4} \right] ({}^t\ddot{\mathbf{U}} + {}^{t+\Delta t/2}\ddot{\mathbf{U}}) \quad (7.3-7)$$

$${}^{t+\Delta t/2}\mathbf{U} = {}^t\mathbf{U} + \left[\frac{\Delta t}{4} \right] ({}^t\dot{\mathbf{U}} + {}^{t+\Delta t/2}\dot{\mathbf{U}}) \quad (7.3-8)$$

and for the second sub-step we use:

$${}^{t+\Delta t}\dot{\mathbf{U}} = \frac{1}{\Delta t} {}^t\mathbf{U} - \frac{4}{\Delta t} {}^{t+\Delta t/2}\mathbf{U} + \frac{3}{\Delta t} {}^{t+\Delta t}\mathbf{U} \quad (7.3-9)$$

$${}^{t+\Delta t}\ddot{\mathbf{U}} = \frac{1}{\Delta t} {}^t\dot{\mathbf{U}} - \frac{4}{\Delta t} {}^{t+\Delta t/2}\dot{\mathbf{U}} + \frac{3}{\Delta t} {}^{t+\Delta t}\dot{\mathbf{U}} \quad (7.3-10)$$

From these equations, we obtain:

$$\left(\frac{16}{\Delta t^2} \mathbf{M} + \frac{4}{\Delta t} \mathbf{C} + \mathbf{K} \right) {}^{t+\Delta t/2}\mathbf{U} = {}^{t+\Delta t/2}\hat{\mathbf{R}} \quad (7.3-11)$$

with

$${}^{t+\Delta t/2}\hat{\mathbf{R}} = {}^{t+\Delta t/2}\mathbf{R} + \mathbf{M} \left(\frac{16}{\Delta t^2} {}^t\mathbf{U} + \frac{8}{\Delta t} {}^t\dot{\mathbf{U}} + {}^t\ddot{\mathbf{U}} \right) + \mathbf{C} \left(\frac{4}{\Delta t} {}^t\mathbf{U} + {}^t\dot{\mathbf{U}} \right) \quad (7.3-12)$$

and

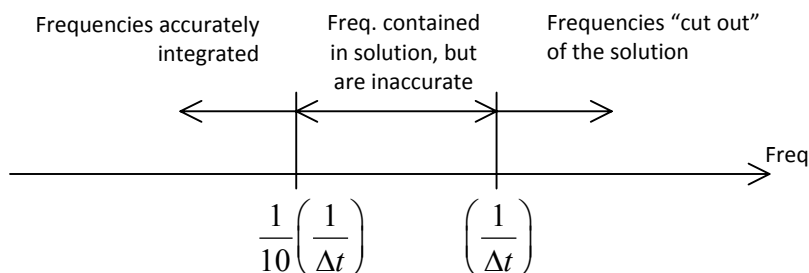
$$\left(\frac{9}{\Delta t^2} \mathbf{M} + \frac{3}{\Delta t} \mathbf{C} + \mathbf{K} \right) {}^{t+\Delta t}\mathbf{U} = {}^{t+\Delta t}\hat{\mathbf{R}} \quad (7.3-13)$$

The complete solution scheme of the Bathe method is given in Ref. KJB, Table 9.4.

- The Bathe method accurately integrates the low frequency modes that are spatially resolved in the finite element model, and automatically damps out (‘cuts out’) the spurious high frequency modes that are not spatially resolved in the model. In essence, the Bathe method acts as a low-pass filter with nearly optimal characteristics. This approach results in remarkably stable and accurate solutions. Some properties of the Bathe method are discussed in

ref. G. Noh, S. Ham and K.J. Bathe, “Performance of an Implicit Time Integration Scheme in the Analysis of Wave Propagations”, *Computers & Structures*, Vol. 123, pp. 93-105, 2013.

- The Bathe method is unconditionally stable, hence the time step size need only be chosen based on accuracy considerations.
- The Bathe method accurately integrates frequencies $f \leq \frac{1}{10} \left(\frac{1}{\Delta t} \right)$, provided the mesh can spatially resolve these frequencies, and automatically damps out frequencies $f > \left(\frac{1}{\Delta t} \right)$, as illustrated below.



- Therefore, if the maximum frequency of interest contained in the solution is f_{\max} , the time step size, Δt , used for the Bathe method should be:

$$\Delta t = \frac{1}{10} \left(\frac{1}{f_{\max}} \right) \quad \text{or} \quad \frac{\Delta t}{T_{\min}} = \frac{1}{10}$$

where T_{\min} is the corresponding smallest period to be integrated.

- In wave propagation problems, it is known that the standard Bathe method performs best with CFL=1.

ref. G. Noh, S. Ham and K.J. Bathe, "*Performance of an implicit time integration scheme in the analysis of wave*"

propagations”, Computers & Structures, 123, 93-105, 2013

- The ρ_∞ -Bathe method is a generalization of the standard Bathe method. The parameter ρ_∞ refers to the spectral radius of the method when the ratio of time step to period is very large. The method has the following features:

When $\rho_\infty = 0.0$, the “standard Bathe method” is obtained. This method might be used primarily.

When $\rho_\infty = 1$, the standard Newmark trapezoidal rule is obtained.

Roughly speaking, for values of ρ_∞ between 0 and 1, the solution characteristics of the ρ_∞ - method are in-between the standard Bathe method and the trapezoidal rule.

- In linear analysis, the time-step is split automatically into two sub-steps by using the formula

$$\gamma_0 = \frac{2 - \sqrt{2 + 2\rho_\infty}}{1 - \rho_\infty} \quad ; \quad \gamma_0 = 0.5 \text{ if } \rho_\infty = 1$$

In this way the optimal sub-step sizes, $\gamma_0\Delta t$ and $(1 - \gamma_0)\Delta t$, are employed in linear analysis which also means that only one coefficient matrix is used and triangularized at the beginning of the time stepping.

- The trapezoidal rule has been employed abundantly in transient analyses, although it displays serious shortcomings. In some analyses accurate results are obtained using the trapezoidal rule, namely when all modes of the finite element system are excited and need be integrated. A small enough time step to integrate accurately the highest frequency need then be chosen. However, frequently only the modes up to a cut-off frequency are accurately represented in the finite element system and only these modes

need be accurately integrated. The rest of the modes should be suppressed and this elimination of the inaccurate modes is achieved automatically when using the Bathe method with ρ_∞ smaller than 1, and frequently most effectively when $\rho_\infty = 0$.

- The following assumptions are used in the ρ_∞ -Bathe method:

For the first sub-step, at time $t + \gamma\Delta t$ we use:

$$\hat{\mathbf{K}}_1^{t+\gamma\Delta t} \mathbf{U} = {}^{t+\gamma\Delta t}\mathbf{R} + \mathbf{M} \left({}^t\ddot{\mathbf{U}} + a_4 {}^t\dot{\mathbf{U}} + a_0 {}^t\mathbf{U} \right) + \mathbf{C} \left({}^t\dot{\mathbf{U}} + a_1 {}^t\mathbf{U} \right) \quad (7.3-14)$$

where

$$\begin{aligned} \hat{\mathbf{K}}_1 &= \mathbf{K} + a_0 \mathbf{M} + a_1 \mathbf{C} \\ a_0 &= \frac{4}{\gamma^2 \Delta t^2} & a_1 &= \frac{2}{\gamma \Delta t} & a_2 &= \frac{1}{\Delta t^2 q_2^2} \\ a_3 &= \frac{1}{\Delta t q_2} & a_4 &= \frac{4}{\gamma \Delta t} & a_5 &= \frac{q_0 + q_2}{\Delta t q_2^2} \\ a_6 &= \frac{q_1}{\Delta t q_2^2} & a_7 &= \frac{q_0}{q_2} & a_8 &= \frac{q_1}{q_2} \end{aligned} \quad (7.3-15)$$

and $\gamma = \gamma_0$.

The method is second order accurate for any ρ_∞ and γ , provided that the following relations are employed:

$$\begin{aligned} q_1 &= \frac{\rho_\infty + 1}{2\gamma(\rho_\infty - 1) + 4} \\ q_0 &= (\gamma - 1)q_1 + \frac{1}{2} \\ q_2 &= -\gamma q_1 + \frac{1}{2} \end{aligned} \quad (7.3-16)$$

After solving for ${}^{t+\gamma\Delta t}\mathbf{U}$, we can calculate:

$$\begin{aligned} {}^{t+\gamma\Delta t}\dot{\mathbf{U}} &= a_1({}^{t+\gamma\Delta t}\mathbf{U} - {}^t\mathbf{U}) - {}^t\dot{\mathbf{U}} \\ {}^{t+\gamma\Delta t}\ddot{\mathbf{U}} &= a_1({}^{t+\gamma\Delta t}\dot{\mathbf{U}} - {}^t\dot{\mathbf{U}}) - {}^t\ddot{\mathbf{U}} \end{aligned}$$

For the second sub-step, at time $t + \Delta t$, we use:

$$\begin{aligned} \hat{\mathbf{K}}_2 {}^{t+\Delta t}\mathbf{U} = & {}^{t+\Delta t}\mathbf{R} + \mathbf{M}(a_2 {}^t\mathbf{U} + a_5 {}^t\dot{\mathbf{U}} + a_6 {}^{t+\gamma\Delta t}\dot{\mathbf{U}}) \\ & + a_7 {}^t\ddot{\mathbf{U}} + a_8 {}^{t+\gamma\Delta t}\ddot{\mathbf{U}} + \mathbf{C}(a_3 {}^t\mathbf{U} + a_7 {}^t\dot{\mathbf{U}} + a_8 {}^{t+\gamma\Delta t}\dot{\mathbf{U}}) \end{aligned} \quad (7.3-17)$$

where

$$\hat{\mathbf{K}}_2 = \mathbf{K} + a_2\mathbf{M} + a_3\mathbf{C}$$

Note that $\hat{\mathbf{K}}_2 = \hat{\mathbf{K}}_1$ if $\gamma = \gamma_0$, hence the effective stiffness matrix needs to be factorized only once for a given time step.

After solving the above equation for displacements at time $t + \Delta t$, we can calculate velocities and accelerations at time $t + \Delta t$:

$$\begin{aligned} {}^{t+\Delta t}\dot{\mathbf{U}} &= -a_3 {}^t\mathbf{U} + a_3 {}^{t+\Delta t}\mathbf{U} - a_7 {}^t\dot{\mathbf{U}} - a_8 {}^{t+\gamma\Delta t}\dot{\mathbf{U}} \\ {}^{t+\Delta t}\ddot{\mathbf{U}} &= -a_3 {}^t\dot{\mathbf{U}} + a_3 {}^{t+\Delta t}\dot{\mathbf{U}} - a_7 {}^t\ddot{\mathbf{U}} - a_8 {}^{t+\gamma\Delta t}\ddot{\mathbf{U}} \end{aligned} \quad (7.3-18)$$

ref. G. Noh, K.J. Bathe, "The Bathe time integration method with controllable spectral radius: The ρ_∞ -Bathe method, *Computers and Structures*, Vol. 212, pp. 299-310 (2019)

ref. G. Noh and K.J. Bathe, "For Direct Time Integrations: A Comparison of the Newmark and ρ_∞ -Bathe schemes", *Computers & Structures*, 225, 2019.

ref. KJB
Sections 9.2.4
and 9.4.4

- The trapezoidal rule (also called the constant-average-

acceleration method of Newmark) is obtained from the Newmark method by using $\delta = 0.5$, $\alpha = 0.25$.

- The trapezoidal rule has the following characteristics:
 - ▶ It is an implicit integration method, meaning that equilibrium of the system is considered at time $t+\Delta t$ to obtain the solution at time $t+\Delta t$.
 - ▶ It is unconditionally stable for linear analysis. Hence, the time step size Δt is selected based on accuracy considerations only, see ref. KJB, Section 9.4.4.

If high frequencies are excited that are spurious (spatially unresolved frequencies) these will not be damped out but appear as spurious ringing in the response.

ref. K. J. Bathe and G. Noh, “Insight into an Implicit Time Integration Scheme for Structural Dynamics”, *Computers & Structures*, 98-99, 1-6, 2012.

ref. Kroyer, K. Nilsson, and K.J. Bathe, “Advances in Direct Time Integration Schemes for Dynamic Analysis”, *automotive CAE Companion 2016/2017*, 32-35, 2016.

- The Newmark method is in general stable when the following conditions are satisfied: $\delta \geq 0.5$, $\alpha \geq 0.25(\delta + 0.5)^2$.
- The step-by-step direct integration solution of a vibration problem is frequently more effective than a mode superposition analysis when the response need only be calculated over relatively short time period, say with less than 1,000 steps.
- The time step increment (Δt) recommended for dynamic analysis with the trapezoidal rule is given by $\omega_{co}\Delta t \leq 0.20$ where ω_{co} is the highest frequency of interest in the dynamic response.
- In FSI analysis with ADINA-FSI, the time integration is performed using the Bathe method for the first and second-order

equations. .

- Whether the mass and damping matrices are diagonal or banded (lumped or consistent discretization), the solution always requires that a coefficient matrix be assembled and factorized.

7.3.2 Time history by mode superposition

ref. KJB
Section 9.3

- In mode superposition analysis, the governing finite element equations are solved using the transformation

$$\mathbf{U} = \sum_{i=r}^s \phi_i x_i$$

where the $\phi_i, i = r, \dots, s$ are the mode shapes calculated in a frequency solution, and the x_i are the corresponding unknown generalized displacements. The x_i are calculated by solving the decoupled modal equations

$$\ddot{x}_i + 2\xi_i \omega_i \dot{x}_i + \omega_i^2 x_i = r_i$$

where ξ_i is the critical damping ratio corresponding to the frequency ω_i , and $r_i = \phi_i^T \mathbf{R}$. The Newmark method (trapezoidal rule) is used for the time integration.

- When \mathbf{R} includes mass-proportional loads, the mass-proportional load calculation does not include the mass coupling between active and deleted degrees of freedom.
- In ADINA, either the mode shapes corresponding to the smallest frequencies or the frequencies in a band can be used (see Chapter 6).
- Using ADINA, either no damping is specified, or modal damping is specified, in which case the values of $\xi_i, i = r, \dots, s$ can be all different. Concentrated dampers are ignored.
- Mode superposition is effective when the time integration has to be carried out over many time steps (e.g., earthquake loading), and the cost of calculating the required frequencies and mode shapes is

reasonable.

- Prescribed displacement loads, prescribed velocities, and prescribed accelerations can be employed in mode superposition analysis.
- Static correction can be applied in linear mode superposition analysis. The static correction is calculated as follows:

$$\Delta \mathbf{R} = \mathbf{R} - \sum_{i=1}^p r_i (\mathbf{M} \cdot \Phi_i)$$

where

p is the number of calculated mode shapes;

$\mathbf{M} \cdot \Phi_i$ is calculated and stored on disk at the beginning of mode superposition solutions;

$r_i = \Phi_i^T \cdot \mathbf{R}$ is calculated during modal solutions and stored for every requested static correction step; and

\mathbf{R} is the external load vector

We then solve

$$\mathbf{K} \cdot \Delta \mathbf{U}(t) = \Delta \mathbf{R}(t)$$

- If static correction is used, static correction responses follow mode superposition responses (e.g., in graphs) for the same time spans. Note that static correction solutions can be requested for a different (smaller) number of steps than the number of steps used in mode superposition solutions.

7.3.3 Lumped and consistent damping; Rayleigh damping

- For a lumped damping matrix, contributions come from concentrated nodal dampers and/or lumped general element dampers and/or lumped spring element dampers. Direct time

integration must be employed.

ref. KJB
Section 9.3.3

- For a consistent damping matrix, contributions come from Rayleigh damping and/or consistent general element damping and/or consistent spring element damping and/or concentrated nodal dampers. Direct time integration must be employed. Explicit time integration cannot be used. Substructuring cannot be used.
- If Rayleigh damping is specified, the contributions of the following matrix ($\mathbf{C}_{Rayleigh}$) are added to the total system damping matrix \mathbf{C} described in Section 7.3:

$$\mathbf{C}_{Rayleigh} = \alpha \mathbf{M} + \beta \mathbf{K}$$

where \mathbf{M} is the total system mass matrix which can be lumped or consistent, and \mathbf{K} is the (initial) total system stiffness matrix (corresponding to zero initial displacements).

- Different Rayleigh damping factors can be input for different element groups.
- Note that $\mathbf{C}_{Rayleigh}$ is in general a consistent damping matrix ($\mathbf{C}_{Rayleigh}$ is diagonal if β is zero and a lumped mass matrix is used).
- If the Rayleigh damping option UPDATE is set to STEP or MASS, then the damping matrix is updated either every time step or when the mass matrix is changed; otherwise, $\mathbf{C}_{Rayleigh}$ is constant throughout the solution and it is formed only once in ADINA – before the step-by-step solution of the equilibrium equations.
- See Ref. KJB, Section 9.3.3, for information about selecting the Rayleigh damping constants α , β . In the modal basis, the damping ratio can be written as

$$\xi_i = \frac{\alpha}{2\omega_i} + \frac{\beta\omega_i}{2}$$

where ξ_i is the damping ratio for mode ω_i . It is seen that α tends to damp lower modes and β tends to damp higher modes.

If α is not used, it is seen that a value of $\beta = \frac{T_p}{\pi}$ will overdamp all motions with periods smaller than T_p . Hence motions with periods smaller than T_p can be suppressed by choosing $\beta = \frac{T_p}{\pi}$.

This may be of interest when using damping to suppress numerical oscillations.

The above comments apply only when the stiffness matrix does not change significantly during the analysis, however.

The above formula and comments do not apply when potential based elements are used.

- If the potential-based fluid elements are used in the model, the matrices **M** and **K** also include the fluid elements contributions (see Section 2.11.2). However, the fluid element contributions to **M** and **K** are not included in $\mathbf{C}_{Rayleigh}$.

7.3.4 Modal damping ratios based on strain energy proportional damping Objective

To enable the user to calculate appropriate values of modal damping ratios ξ_i in transient modal analysis, based on strain energy proportional damping ratios h_i . These damping ratios can be either constant over the whole structure, or locally varying between components (element groups).

In damped modal analysis in ADINA, the following system of equations is solved

$$\ddot{x}_i(t) + 2\omega_i\xi_i\dot{x}_i(t) + \omega_i^2x_i(t) = r_i(t) \quad i = 1, 2, \dots, p$$

where i is the mode number, ranging from 1 to p , ω_i is the natural

frequency and ξ_i is the *modal damping ratio*. The modal equations are all decoupled. For more details, see Section 7.3.2 of the ADINA-TMG, or Section 9.3.3 of the Finite Element Procedures book.

If the user requires a different form of damping, such as strain energy damping, then that damping form must be converted to “equivalent” modal damping ratios. A standard damped modal analysis is then performed.

Two types of strain energy proportional damping will be allowed by ADINA. In the first type, *energy damping ratios* applicable to the whole structure are input. In the second type, the energy damping ratios are allowed to vary between different element groups.

Global strain energy damping

The energy damping ratio h_i for strain energy proportional damping is defined for each mode i as

$$h_i = \frac{\Delta E_i}{E_i} \quad (7.3-19)$$

where ΔE_i is the energy dissipated in the i -th mode, and E_i is the total strain energy for that mode. In axisymmetric analysis, this energy damping ratio provides the energy dissipated per unit radian.

The total strain energy for mode i in modal analysis is proportional to the projected stiffness matrix

$$E_i = \frac{1}{2} \boldsymbol{\phi}_i^T \mathbf{K} \boldsymbol{\phi}_i = \frac{1}{2} \omega_i^2 ; \quad \boldsymbol{\phi}_i^T \mathbf{M} \boldsymbol{\phi}_i = 1 \quad (7.3-20)$$

where $\boldsymbol{\phi}_i$ is mode shape i based on an undamped system and \mathbf{K} and \mathbf{M} are the global stiffness and mass matrices.

Local strain energy damping

If the strain energy damping ratios are not uniform throughout the

structure, ADINA will allow each sub-structure or component (represented as element group in ADINA) to have its own *local energy damping ratios*

$$h_{i|j} = \frac{\Delta E_{i|j}}{E_{i|j}} \quad (7.3-21)$$

where $\Delta E_{i|j}$ and $E_{i|j}$ are the dissipated and total strain energy for mode i for element group j .

The strain energies in Eqn. (7.3-21) can be expressed as

$$E_{i|j} = \frac{1}{2} \boldsymbol{\phi}_i^T \mathbf{K}_j \boldsymbol{\phi}_i$$

where $\boldsymbol{\phi}_i$ is undamped mode shape i for the nodes in element group j and \mathbf{K}_j is the stiffness matrix for element group j . $\boldsymbol{\phi}_i$ is a subset of mode shape i for the whole structure.

ADINA allows each element group to have its own input of local energy damping ratios $h_{i|j}$ and then calculates the global energy damping ratios using

$$h_i = \frac{\sum_{j=1}^N E_{i|j} h_{i|j}}{\sum_{j=1}^N E_{i|j}}$$

where N is the number of element groups. This calculation is performed before the damped modal analysis, using the mode shapes

$$h_i = \frac{\sum_{j=1}^N \boldsymbol{\phi}_i^T \mathbf{K}_j \boldsymbol{\phi}_i h_{i|j}}{\sum_{j=1}^N \boldsymbol{\phi}_i^T \mathbf{K}_j \boldsymbol{\phi}_i} = \frac{1}{\omega_i^2} \sum_{j=1}^N \boldsymbol{\phi}_i^T \mathbf{K}_j \boldsymbol{\phi}_i h_{i|j} \quad (7.3-22)$$

Once the global strain energy damping ratios h_i are available, the

modal damping ratio for each mode is set by ADINA as

$$\xi_i = \frac{h_i}{2} \quad (7.3-23)$$

A standard damped modal analysis is then performed.

Notes

It is the user's responsibility to provide proper values of strain energy damping ratios, h_i in the case of global parameters, or h_{ij} in case of local damping.

In the case of local strain energy damping, the AUI will provide means for the user to apply the damping ratios to multiple element groups simultaneously, instead of each element group separately.

Equation (7.3-16) is applicable for small damping ratios. It can be derived, for a single degree of freedom model, by comparing the rate of decay resulting from the damping ratio ξ_i , and that from the strain energy damping ratio h_i . For viscous damping, the following relationship holds for small amount of damping

$$\xi = \frac{\delta}{\sqrt{\delta^2 + 4\pi^2}} \quad (7.3-24)$$

where δ is the logarithmic decrement

$$\delta = \ln\left(\frac{x_0}{x_1}\right)$$

which can be expressed in terms of the strain energy damping ratio, as follows

$$\delta = -\frac{1}{2} \ln\left(\frac{E_1}{E_0}\right) = -\frac{1}{2} \ln\left(\frac{E_0(1-h)^{2\pi}}{E_0}\right) = -\pi \ln(1-h) \quad (7.3-25)$$

Substituting Eqn. (7.3-25) into Eqn. (7.3-24) and assuming small values of ζ and h , results in Eqn. (7.3-23).

AUI Input

Modal damping ratios are calculated if one of the modal analysis options is specified (e.g., MODAL-TRANSIENT, MODAL-PARTICIPATION-FACTORS, etc.) and if the following command is used:

MODAL-DAMPING OPTION=LOCAL EGROU=...

This command can be used for every element group. If EGROU = 0 is specified then the damping factors are taken for all groups not listed in the MODAL-DAMPING command input. Note that if MODAL-TRANSIENT analysis is specified, then modal damping factors are taken according to Eqn. (7.3-11).

Modal damping ratios are listed in the output (.out) file.

Ref. K. Kawashima, H. Nagashima and H. Iwasaki,
"Evaluation of Modal Damping Ratio Based on Strain
Energy Proportional Damping Method", *Journal of
Structural Engineering*, Vol 40A, 1994.

7.4 Nonlinear dynamic analysis

ref. KJB
Section 9.5

- In nonlinear dynamic analysis the solution of the finite element equations is usually obtained by direct integration procedures. The following major techniques can be employed with ADINA:

Explicit integration using the central difference method or the Noh-Bathe method (described in Chapter 8)

Implicit integration using the Newmark method or the Bathe method

Some considerations on the choice of the time integration method are given in Section 7.5.

- For certain types of problems it can be effective to use mode superposition for the solution of the nonlinear response. In this case, the finite element equilibrium equations are established in the selected mode shapes and solved by using the Newmark method (see Section 7.4.2).
- It can also be effective to use substructuring for the solution of the equations (see Section 10.1).
- The use of Rayleigh damping is the same as described in Section 7.3.3. Note that the Rayleigh damping matrix remains constant throughout the time integration if option UPDATE=none is used in the RAYLEIGH-DAMPING command. If UPDATE=step or mass is used, then the damping matrix is updated according to the above requests.

7.4.1 Step-by-step implicit time integration

- In nonlinear analysis the incremental finite element equilibrium equations used are

$$\mathbf{M}^{t+\Delta t} \ddot{\mathbf{U}}^{(i)} + \mathbf{C}^{t+\Delta t} \dot{\mathbf{U}}^{(i)} + {}^{t+\Delta t}\mathbf{K} \Delta \mathbf{U}^{(i)} = {}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F}^{(i-1)} \quad (7.4-2)$$

where ${}^{t+\Delta t}\ddot{\mathbf{U}}^{(i)}$, ${}^{t+\Delta t}\dot{\mathbf{U}}^{(i)}$, ${}^{t+\Delta t}\mathbf{U}^{(i-1)} + \Delta \mathbf{U}^{(i)}$ are the approximations to the accelerations, velocities, and displacements obtained in iteration (i) respectively.

The vector of nodal point forces ${}^{t+\Delta t}\mathbf{F}^{(i-1)}$ is equivalent to the element stresses in the configuration corresponding to the displacements ${}^{t+\Delta t}\mathbf{U}^{(i-1)}$.

- The **Newmark method** is summarized in Chapter 7.3.1. . The trapezoidal rule is obtained by using $\delta = 0.5$ and $\alpha = 0.25$ and is recommended if the Newmark method is used.
- The **Bathe method** is summarized in Chapter 7.3.1.

- For nonlinear solutions, new coefficient matrices are formed throughout the solution and hence the value γ is a user input parameter with a default value equal to 0.5. Note that for linear solutions γ is set to $2 - \sqrt{2}$ and cannot be overwritten by an input.
- In nonlinear analysis, , in particular, in problems with large displacements or problems involving contact, the Newmark method can become unstable, while the Bathe method remains stable and accurate, hence, a much larger time step size can often be used with the Bathe method. Guidelines on selecting the time step size for the Bathe method are given in Chapter 7.3.1.
- The dynamic equilibrium equations are solved by use of the iterative procedures also used in static analysis, including the ATS method. Line search is also applicable to dynamic analysis. However, the LDC method cannot be used (see Chapter 7.2.1).
- The energy and force/moment convergence criteria used in nonlinear dynamic analysis are:

Energy convergence criterion: For all degrees of freedom

$$\frac{\Delta \mathbf{U}^{(i)T} \left[{}^{t+\Delta t} \mathbf{R} - \mathbf{M} {}^{t+\Delta t} \ddot{\mathbf{U}}^{(i-1)} - \mathbf{C} {}^{t+\Delta t} \dot{\mathbf{U}}^{(i-1)} - {}^{t+\Delta t} \mathbf{F}^{(i-1)} \right]}{\Delta \mathbf{U}^{(1)T} \left[{}^{t+\Delta t} \mathbf{R} - \mathbf{M} {}^{t+\Delta t} \ddot{\mathbf{U}}^{(0)} - \mathbf{C} {}^{t+\Delta t} \dot{\mathbf{U}}^{(0)} - {}^{t+\Delta t} \mathbf{F} \right]} \leq \text{ETOL}$$

Force and moment convergence criteria: For the translational degrees of freedom

$$\frac{\left\| {}^{t+\Delta t} \mathbf{R} - \mathbf{M} {}^{t+\Delta t} \ddot{\mathbf{U}}^{(i-1)} - \mathbf{C} {}^{t+\Delta t} \dot{\mathbf{U}}^{(i-1)} - {}^{t+\Delta t} \mathbf{F}^{(i-1)} \right\|_2}{\text{RNORM}} \leq \text{RTOL}$$

For the rotational degrees of freedom

$$\frac{\left\| {}^{t+\Delta t}\mathbf{R} - \mathbf{M} {}^{t+\Delta t}\dot{\mathbf{U}}^{(i-1)} - \mathbf{C} {}^{t+\Delta t}\dot{\mathbf{U}}^{(i-1)} - {}^{t+\Delta t}\mathbf{F}^{(i-1)} \right\|_2}{\text{RMNORM}} \leq \text{RTOL}$$

The other convergence criteria and the notation and considerations for the use of the convergence criteria are the same as in nonlinear static analysis; see Chapter 7.2.2.

- Time functions can be applied the global mass matrix and global damping matrix in step-by-step implicit time integration. Applying a time function to the global mass matrix and global damping matrix scales the inertia forces and damping forces as a function of time, respectively. This can be used to control when the inertia forces and/or damping forces are applied in the analysis.
- Applying a time function to the global damping matrix does not scale the damping forces of nonlinear spring elements and connector elements, as these elements do not contribute to the global damping matrix, instead the damping effect is added to the out-of-balance force vector and the global stiffness matrix.
- Note that applying a time function to the global damping matrix does scale the damping forces of linear spring elements, as these elements contribute to the global damping matrix.
- Time functions cannot be applied to the global mass matrix and global damping matrix when there are potential-based elements in the model.

Mass-proportional loading, and centrifugal/rotational loading is not scaled by the time function applied to the global mass matrix.

7.4.2 Time history by mode superposition

ref. KJB
Section 9.5.3

- A nonlinear dynamic problem is solved effectively by the mode superposition procedure if the response can be adequately represented by the superposition of the response in a few selected mode shapes.
- The mode shapes result from a linearization about a reference configuration and are not updated during the mode superposition

analysis. The reference configuration can be based on a restart analysis.

- The mode shapes can be calculated at the start of the nonlinear mode superposition analysis or read from a prior frequency analysis file.
- The equations solved in ADINA to obtain the solution of the nonlinear dynamic equations by mode superposition are presented on pp. 828-829 of ref. KJB. In ADINA, throughout the time integration, the eigenmodes corresponding to the time at solution start are employed, and the modal equations are integrated using the Newmark method.
- If desired, nonlinear mode superposition analysis can be performed in stages. Each restart stage can start by recalculating the eigenmodes based on the last available solution.
- If the model during the mode superposition analysis experiences significant further nonlinearity the mode shapes of the reference configuration may no longer be suitable. This happens, for example, if more nodes come into or go out of contact, or if frictional states change between stick and slip. For such problems, a non-modal nonlinear dynamic analysis should be performed.
- It is the user's responsibility to ensure that sufficient mode shapes are used, and that they are applicable throughout the nonlinear dynamic simulation.
- Some applications using the mode superposition method are given in the following reference:

ref. K.J. Bathe and S. Gracewski, "On Nonlinear Dynamic Analysis Using Substructuring and Mode Superposition," *J. Computers and Structures*, Vol. 13, pp. 699-707, 1981.

7.4.3 Global mass matrix

- The global mass matrix used in nonlinear dynamic analysis is the same as the global mass matrix used in linear dynamic analysis.

- Unlike the global stiffness matrix, which is typically recalculated during each equilibrium iteration, the global mass matrix is recalculated only for the following case:

Element birth-death: only the elements that are currently alive contribute to the global mass matrix. (This includes the case of elements that have ruptured; ruptured elements also do not contribute to the global mass matrix.)

- In the following discussion, we consider an analysis without element birth-death. In this case, the global mass matrix is calculated only once, at the beginning of the analysis.
- For continuum elements, in which the unknowns are the displacement degrees of freedom, the mass matrix does not change due to large deformations (ref KJB, p 542). Hence, in this case, it is theoretically justified to not recalculate the global mass matrix.
- However, it should be recognized that there are some situations in which, theoretically, the global mass matrix changes during the analysis. These situations are all associated with rotational degrees of freedom.

Structural elements with rotational degrees of freedom, consistent mass matrix. The effect is most pronounced when the element cross-section or thickness is large, and is also most pronounced for the Hermitian beam element.

Rigid links and constraint equations with rotational degrees of freedom

Concentrated rotational masses with different inertias in different rotational directions

- Typically the effect of neglecting the time-varying parts of the global mass matrix is small. However, for certain analyses in which there are large rotations, the effect can be significant.
- A time function can be applied to the global mass matrix to scale the inertia force as a function of time, as described in Section

7.4.1.

7.5 Choosing between implicit and explicit formulations

*ref. KJB
Section 9.2*

- The main criterion governing the selection of the implicit or explicit formulations is the time scale of the solution.
- The implicit method can use much larger time steps since it is unconditionally stable. However, it involves the assembly and solution of a system of equations, and it is iterative. Therefore, the computational time per load step is relatively high. The explicit method uses much smaller time steps since it is conditionally stable, meaning that the time step for the solution has to be less than a certain critical time step, which depends on the smallest element size and the material properties. However, it involves no matrix solution and is non-iterative. Therefore, the computational time per load step is relatively low.
- For both linear and nonlinear static problems, the implicit method is the only option.
- For low-speed dynamic problems, the solution time spans a period of time considerably longer than the time it takes the wave to propagate through an element. The solution in this case is dominated by the lower frequencies of the structure. This class of problems covers most structural dynamics problems, certain metal forming problems, crush analysis, earthquake response and biomedical problems. When the explicit method is used for such problems the resulting number of time steps will be excessive, unless mass-scaling is applied, or the loads are artificially applied over a shorter time frame. No such modifications are needed in the implicit method. Hence, the implicit method is the optimal choice.
- For high-speed dynamic problems, the solution time is comparable to the time required for the wave to propagate through the structure. This class of problems covers most wave propagation problems, explosives problems, and high-speed impact problems. For these problems, the number of steps required with the explicit method is not excessive. If the implicit method uses a similar time step it will be much slower and if it uses a much larger time step it will introduce other solution errors since it will not be capturing the

pertinent features of the solution (but it will remain stable). Hence, the explicit method is the optimal choice.

- A large number of dynamics problems cannot be fully classified as either low-speed or high-speed dynamic. This includes many crash problems, drop tests and metal forming problems. For these problems both solution methods are comparable. However, whenever possible (when the time step is relatively large and there are no convergence difficulties) we recommend the use of the implicit solution method.
- Note that the explicit solution provided in ADINA does not use reduced integration with hour-glassing. This technique reduces the computational time per load step. However, it can have detrimental effect on the accuracy and reliability of the solution.
- Since the explicit time step size depends on the length of the smallest element, one excessively small element will reduce the stable time step for the whole model. Mass-scaling can be applied to these small elements to increase their stable time step. The implicit method is not sensitive to such small elements.
- Since the explicit time step size depends on the material properties, a nearly incompressible material will also significantly reduce the stable time step. The compressibility of the material can be increased in explicit analysis to achieve a more acceptable solution time. The implicit method is not as sensitive to highly incompressible materials (provided that a mixed formulation is used).
- Higher order elements such as the 10-node tetrahedral, 20 and 27 node brick elements are only available in implicit analysis. They are not used in explicit analysis because no suitable mass-lumping technique is available for these elements.
- Model nonlinearity is another criterion influencing the choice between implicit and explicit solutions. As the level of nonlinearity increases, the implicit method requires more time steps, and each time step may require more iterations to converge. In some cases, no convergence is reached. The explicit method however, is less sensitive to the level of nonlinearity.

Note that when the implicit method fails it is usually due to non-convergence within a time step, while when the explicit method fails it is usually due to a diverging solution.

- The memory requirements is another factor. For the same mesh, the explicit method requires less memory since it does not store a stiffness matrix and does not require a solver. This can be significant for very large problems.
- Since ADINA handles both implicit and explicit analysis with very similar inputs, the user can in many cases restart from one analysis type to the other. This capability can be used, for example, to perform implicit springback analysis following an explicit metal forming simulation, or to perform an explicit analysis following the implicit application of a gravity load.

It can also be used to overcome certain convergence difficulties in implicit analyses. A restart from the last converged implicit solution to explicit can be performed, then, once that stage is passed, another restart from explicit to implicit can be performed to proceed with the rest of the solution.

7.6 Tracking solution progress

- Important model parameters such as the memory used by the model, memory used by the solver, number of degrees of freedom, solution times, warning messages, and error messages are all provided in the .out file.
- Detailed iteration by iteration convergence information is also written to the .out file as illustrated in the example of Section 7.2.9.
- The program outputs a more summarized time step information to the .msg file. This output focuses on the time steps and the ATS history.

7.6.1 .rto file

- Several solution control parameters can be modified during execution by creating a runtime option file with extension .rto. The .rto file can be created either “by hand”, using the text editor of

your choice, or by the AUI, using the RTOFILE command.

- The syntax of the .rto file is now described.

The .rto file can contain an arbitrary number of lines. Each line in the .rto file is a string of up to 80 characters. A line beginning with a * or # is a comment line. Any text following // is a comment.

All non-comment lines or text are converted to upper case.

Each line of the .rto file sets a parameter. Parameters are set using syntax such as

```
DTOL=0.234
ETOL=1E-5
```

Allowable parameters are given in Table 7-6.

- ADINA reads the .rto file at the start of program execution, and also before each equilibrium iteration. It is allowed to modify the .rto file during program execution, for example to reset the convergence tolerances during equilibrium iterations, or to save the current solution, then stop. Use your favorite text editor to modify the .rto file.

Table 7-6: Parameters that can be set in the .rto file

Parameter name	Description	Allowable values
DMNORM	Reference value for displacement norm, rotations	Real, > 0.0
DNORM	Reference value for displacement norm, translations	Real, > 0.0
DTOL	Tolerance for displacement norm	Real, > 0.0
ETOL	Tolerance for energy norm	Real, > 0.0

Parameter name	Description	Allowable values
MAXDISP	Limiting incremental displacement during equilibrium iterations	Real, > 0.0
MAXITE	Maximum number of iterations allowed during equilibrium iterations	Integer, > 0
MINITE	Minimum number of iterations performed during equilibrium iterations	Integer, > 0
RCONSM	Reference contact force level	Real, > 0.0
RCTOL	Contact force convergence tolerance	Real, > 0.0
RMNORM	Reference value for force norm, moments	Real, > 0.0
RNORM	Reference value for force norm, forces	Real, > 0.0
RTOL	Tolerance for force norm	Real, > 0.0
STOL	Tolerance for line searching	Real, > 0.0
STOP	Saves the previously converged solution in the porthole file, then stops program execution.	Either the string 'YES' or the integer '1'. E.g., STOP=YES, or STOP=1

7.7 Solution graphs

- ADINA supports the option to plot solution graphs in the utility Graph Viewer. The AUI command GRAPH is used to pre-define the variable list, the results of which can be output to the ITE file. The curves of the results vs solution time/steps can be plotted during runtime or after the solution is complete.
- Multiple graphs can be defined in a model, and up to 12 variables can be requested for each graph. Currently, variable

categories listed in Table 7.7-1 are supported in the Solution Graph, only for static or implicit dynamic analysis.

- Three major variable categories are currently supported for Solution Graph, as listed in the first column of the Table 7-7-1: Node, Element (only for 2D/3D solid elements currently) and Contact. For each major category, there're numbers of sub-categories listed, and more than one variable names could be included in each sub-category, but they're not listed here. For example, for the variable category Displacement, there're seven variable names included: X-Displacement, Y-Displacement, Z-Displacement, X-Rotation, Y-Rotation, Z-Rotation and Displacement_Magnitude. For the full list of all variable names, please see the variable list after the command GRAPH in ADINA command reference manual.
- The variables in the categories of Node and Contact are classified to nodal results, i.e., the results of the variables are collected only from nodes. The variables in the category of Element are classified to the element results, i.e., the results of the variables are collected only from elements. For the results collected from more than one nodes or elements, eight statistical methods are available to calculate the results: Average, Abs_Average, Sum, Abs_Sum, Max, Abs_Max, Min and Abs_Min.
- The nodal results (variables of Node and Contact categories) are directly collected from all the specified nodes, and then calculated with the statistical method specified. Similarly, the volume/mass/density results of Element category are directly collected from the specified elements, and then calculated with the statistical method specified. However, the stress/strain results have to be collected from each integration points of the element, and then calculated based on the statistical method specified.

Table 7.7-1: Solution graph variables

Variable Category	Variable type		Input type		
	Nodal	Element	Direct	Location	Model
Node					
Displacement	√		√	√	√
Velocity	√		√	√	√
Acceleration	√		√	√	√
Reaction	√		√	√	√
FSI force	√		√	√	√
Reaction torque	√		√	√	√
FSI force torque	√		√	√	√
Prescribed load	√		√	√	√
Prescribed displacement	√		√	√	√
Nodal position	√		√	√	√
Element (2D/3D solid)					
Element Volume		√	√	√	√
Element Mass		√	√	√	√
Element Mass Density		√	√	√	√
Element Stress		√	√	√	
Element Strain		√	√	√	
Contact					
Nodal Sliding Distance	√		√	√	√
Nodal Slip Velocity	√		√	√	√
Nodal Contact Area	√		√	√	√
Contact Force	√		√	√	√
Contact Traction	√		√	√	√
Contact Status	√		√	√	√

- There are three input-types to define the nodes/elements where to collect the solution results: Direct, Location and Model:
 - ▶ For the Direct method, only one node or element is specified as the location to collect the results.
 - ▶ For the Location method, multiple nodes or elements can be specified through other entities as defined in the command LOCATION. For example, specify a body number in the command LOCATION, and then specify the location number in the command GRAPH, so that the nodal results on all the nodes associated to the body can be requested.
 - ▶ For the Model method, the results are collected from all the applicable nodes/elements in the model. For example, set the variable name as X-Reaction, set the Input-type as Model and set the Method as Summary to request the total reaction along X-direction of the model. The Model method currently does not apply for element stress/strain results.

8. Explicit dynamic analysis

This chapter presents the formulations and algorithms used to solve explicit dynamic problems, including time step calculation. A summary of the elements and material properties available for explicit analysis are listed in Table 8-1. More details are provided in the appropriate sections in Chapters 2 and 3.

Table 8-1: Elements and materials available in explicit time integration

Element type	Nodes	Elastic	Nonlin.-elast.	Elastic-orth.	Drucker-Prag.	Ilyushin	Plastic-bilin.	Plastic-mult.	Plastic-cyclic	Plastic-orth.	Gurson-plast.	Thermo-iso.	Thermo-orth.	Thermo-plast.	Rubber-like ***	Fluid	Gasket**	Concrete & DF Concrete
Truss	2	✓	✓				✓	✓	✓			✓		✓				
Beam	2	✓					✓		✓									
Isobeam	2	✓					✓	✓				✓		✓				
Plate	3	✓		✓		✓												
Shell (MITC)	3/4/9	✓		✓			✓	✓	✓	✓		✓	✓	✓				
3D-Shell	3/4	✓							✓						✓			
Solid (2D, displ.)	3/4/6/9	✓		✓	✓		✓	✓	✓	✓		✓	✓	✓	✓		✓	✓
Solid (2D, u/p)	3/4/6/9	✓		✓			✓	✓	✓	✓		✓	✓	✓	✓			✓
Solid (2D, inc.mod)	3*/4	✓		✓			✓	✓	✓	✓		✓	✓	✓	✓			
Solid (3D, displ.)	4/5/6/8 /10/27	✓		✓	✓		✓	✓	✓	✓		✓	✓	✓	✓		✓	✓
Solid (3D, u/p)	4/5/6/8 /10/27	✓		✓			✓	✓	✓	✓		✓	✓	✓	✓			✓
Solid (3D, inc.mod)	4*/8	✓		✓			✓	✓	✓	✓		✓	✓	✓	✓			
Disp.fluid 2-D	4															✓		
Disp.fluid 3-D	8																	

In addition, linear and nonlinear spring elements are available.

*) 3-node triangular 2-D and 4-node 3-D tetrahedral elements do not use incompatible modes.

**) No critical time step checking for gasket materials

***) The rubber-like materials supported in explicit analysis are the Mooney-Rivlin, Ogden, Arruda-Boyce, eight-chain and Sussman-Bathe materials. Viscoelastic, Mullins and orthotropic effects are not supported.

Table 8-2 lists more element types not available in explicit analysis.

Table 8-2: Elements types not available in explicit analysis

alignment elements
connector elements
potential-based fluid elements
porous media
pipe elements
general elements
user-supplied elements
shell elements not classified as new MITC elements (e.g., layered shells, shells with composite failure criteria, MITC8, isoparametric variable node shells, etc.)

Table 8-3 lists the ADINA options not available in explicit analysis.

Table 8-3: ADINA options not available in explicit analysis

cyclic symmetry
substructuring
mapping
zooming
Thermo-mechanical coupling (TMC)
Fluid-structure interaction (FSI)
consistent mass matrix
consistent Rayleigh damping using the central difference method
temperatures provided from file
load penetration (for ruptured elements)
general constraints
mesh glueing

8.1 Formulation

One of the two following explicit time integration methods can be

used:

- The central difference method (CDM - see ref. KJB, Section 9.2.1)

- The Noh-Bathe method (see reference below)

ref G. Noh and K.J. Bathe, "An explicit time integration scheme for the analysis of wave propagations", *Computers & Structures*, 2013

8.1.1 Central difference method (CDM)

$${}^t\ddot{\mathbf{U}} = \frac{1}{\Delta t^2} \left({}^{t-\Delta t}\mathbf{U} - 2 {}^t\mathbf{U} + {}^{t+\Delta t}\mathbf{U} \right) \quad (8.1-1)$$

and the velocity is calculated using

$${}^t\dot{\mathbf{U}} = \frac{1}{2\Delta t} \left(-{}^{t-\Delta t}\mathbf{U} + {}^{t+\Delta t}\mathbf{U} \right) \quad (8.1-2)$$

The governing equilibrium equation at time t is given by

$$\mathbf{M} {}^t\ddot{\mathbf{U}} + \mathbf{C} {}^t\dot{\mathbf{U}} = {}^t\mathbf{R} - {}^t\mathbf{F} \quad (8.1-3)$$

Substituting the relations for ${}^t\ddot{\mathbf{U}}$ and ${}^t\dot{\mathbf{U}}$ in Eq. (8.1-1) and (8.1-2), respectively, into Eq. (8.1-3), we obtain

$$\left(\frac{1}{\Delta t^2} \mathbf{M} + \frac{1}{2\Delta t} \mathbf{C} \right) {}^{t+\Delta t}\mathbf{U} = {}^t\mathbf{R} - {}^t\mathbf{F} + \frac{2}{\Delta t^2} \mathbf{M} {}^t\mathbf{U} - \left(\frac{1}{\Delta t^2} \mathbf{M} + \frac{1}{2\Delta t} \mathbf{C} \right) {}^{t-\Delta t}\mathbf{U} \quad (8.1-4)$$

from which we can solve for ${}^{t+\Delta t}\mathbf{U}$.

- ref. KJB • The central difference method has the following characteristics:
Sections 9.2.1,
9.4 and 9.5.1
- It is an explicit integration method, meaning that equilibrium of the finite element system is considered at time t to obtain the solution at time $t+\Delta t$.

- ▶ When the mass and damping matrices are diagonal, no coefficient matrix needs to be factorized, see ref. KJB, p. 772. The use of the central difference method is only effective when this condition is satisfied. Therefore, only lumped mass can be used. Also damping can only be mass-proportional.
- ▶ No degree of freedom should have zero mass. This will lead to a singularity in the calculation of displacements according to Eq. 8.1-4, and will also result in a zero stable time step.
- ▶ The central difference method is conditionally stable. The time step size Δt is governed by the following criterion

$$\Delta t \leq \Delta t_{CR} = \frac{T_{Nmin}}{\pi}$$

where Δt_{CR} is the critical time step size, and T_{Nmin} is the smallest period in the finite element mesh.

- The central difference method is most effective when low-order elements are employed. Hence quadratic 3-D solid and shell elements are not allowed.
- The time step can be specified by the user, or calculated automatically. When the user specifies the time, the program does not perform any stability checking. It is the user's responsibility, in this case, to ensure that an appropriate stable time step is used.
- When automatic time step calculation is selected, the TIMESTEP entry is only used to determine the number of nominal time steps and the frequency of output of results. The stable time step is used instead of the value in TIMESTEP (unless the value in TIMESTEP is smaller).

For example, if the user requests 12 steps of size 1.0 with porthole output every 4 steps, there will be 12 nominal steps each of size 1.0. If the stable time step is smaller than 1.0 it will be used instead and results will be saved as soon as the solution time exceeds 4.0, 8.0 and exactly at 12.0 since it is the last step of the analysis.

8.1.2 Noh-Bathe method

- The Noh-Bathe method is a second order accurate explicit time integration scheme, which can be used with or without stiffness and mass based Rayleigh damping. The method is stable for time steps

$$\Delta t \leq \frac{\Delta t_{cr}}{p}$$

where Δt_{cr} is define the same way as for the CDM method and p is a parameter, $p \in (0,1)$. For a given time step Δt , the program uses two sub-steps with time step sizes $p\Delta t$ and $(1-p)\Delta t$.

- For the first sub-step we use the following procedure:

- we calculate displacements and effective loads at time $t + p\Delta t$:

$${}^{t+p\Delta t}\mathbf{U} = {}^t\mathbf{U} + a_0 {}^t\dot{\mathbf{U}} + a_1 {}^t\ddot{\mathbf{U}}$$

$${}^{t+p\Delta t}\widehat{\mathbf{R}} = {}^{t+p\Delta t}\mathbf{R} - {}^{t+p\Delta t}\mathbf{F} - \mathbf{C}({}^t\dot{\mathbf{U}} + a_0 {}^t\ddot{\mathbf{U}})$$

where ${}^{t+p\Delta t}\mathbf{F}$ is an element internal force vector.

- we then solve for accelerations at time $t + p\Delta t$:

$$\mathbf{M} {}^{t+p\Delta t}\ddot{\mathbf{U}} = {}^{t+p\Delta t}\widehat{\mathbf{R}}$$

and we can calculate velocities at time $t + p\Delta t$:

$${}^{t+p\Delta t}\dot{\mathbf{U}} = {}^t\dot{\mathbf{U}} + a_2 ({}^t\ddot{\mathbf{U}} + {}^{t+p\Delta t}\ddot{\mathbf{U}})$$

- For the second sub-step, we calculate displacements for time $t + \Delta t$ and the corresponding load vector:

$$\begin{aligned} {}^{t+\Delta t}\mathbf{U} &= {}^{t+p\Delta t}\mathbf{U} + a_3 {}^{t+p\Delta t}\dot{\mathbf{U}} + a_4 {}^{t+p\Delta t}\ddot{\mathbf{U}} \\ {}^{t+\Delta t}\widehat{\mathbf{R}} &= {}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F} - \mathbf{C}({}^{t+p\Delta t}\dot{\mathbf{U}} + a_3 {}^{t+p\Delta t}\ddot{\mathbf{U}}) \end{aligned}$$

- we can then solve for accelerations at time $t + \Delta t$:

$$\mathbf{M}^{t+\Delta t} \ddot{\mathbf{U}} = {}^{t+\Delta t} \widehat{\mathbf{R}}$$

and we can calculate velocities at $t + \Delta t$:

$${}^{t+\Delta t} \dot{\mathbf{U}} = {}^{t+p\Delta t} \dot{\mathbf{U}} + a_3 {}^t \ddot{\mathbf{U}} + a_6 {}^{t+p\Delta t} \ddot{\mathbf{U}} + a_7 {}^{t+\Delta t} \ddot{\mathbf{U}}$$

where:

$$\begin{aligned} a_0 &= p\Delta t & a_3 &= (1-p)\Delta t \\ a_1 &= \frac{1}{2}(p\Delta t)^2 & a_4 &= \frac{1}{2}((1-p)\Delta t)^2 \\ a_2 &= \frac{1}{2}a_0 & a_5 &= q_0a_3 \end{aligned}$$

$$\begin{aligned} a_6 &= (\frac{1}{2} + q_1)a_3 \\ a_7 &= q_2a_3 \end{aligned}$$

where

$$\begin{aligned} q_1 &= \frac{1-2p}{2p(1-p)} \\ q_2 &= \frac{1}{2} - pq_1 \\ q_0 &= -q_1 - q_2 + \frac{1}{2} \end{aligned}$$

- The external force vector at the first sub-step ${}^{t+p\Delta t} \mathbf{R}$ is evaluated either by interpolation:

$${}^{t+p\Delta t} \mathbf{R} = (1-p) {}^t \mathbf{R} + p {}^{t+\Delta t} \mathbf{R}$$

or by direct evaluation using time functions:

$${}^{t+p\Delta t} \mathbf{R} = \mathbf{R}(t + p\Delta t)$$

- In some cases (e.g., when post-impact calculations are used for the kinematic constraint contact algorithm or if prescribed velocities are used), we need to calculate velocities from displacements and accelerations from velocities:

- for $t + p\Delta t$ step we have:

$${}^{t+p\Delta t}\dot{\mathbf{U}} = a_8({}^{t+p\Delta t}\mathbf{U} - {}^t\mathbf{U}) - {}^t\dot{\mathbf{U}}$$

$${}^{t+p\Delta t}\ddot{\mathbf{U}} = a_8({}^{t+p\Delta t}\dot{\mathbf{U}} - {}^t\dot{\mathbf{U}}) - {}^t\ddot{\mathbf{U}}$$

- for $t + \Delta t$ step we have:

$${}^{t+\Delta t}\dot{\mathbf{U}} = a_9({}^{t+\Delta t}\mathbf{U} - {}^{t+p\Delta t}\mathbf{U}) - {}^{t+p\Delta t}\dot{\mathbf{U}}$$

$${}^{t+\Delta t}\ddot{\mathbf{U}} = a_9({}^{t+\Delta t}\dot{\mathbf{U}} - {}^{t+p\Delta t}\dot{\mathbf{U}}) - {}^{t+p\Delta t}\ddot{\mathbf{U}}$$

where

$$a_8 = \frac{2}{p\Delta t}$$

$$a_9 = \frac{2}{(1-p)\Delta t}$$

8.1.3 Mass matrix

- The construction of the lumped mass matrix depends on the type of element used. Details are provided in the appropriate section in Chapter 2.

For elements with translational degrees of freedom only, the total mass of the element is divided equally among its nodes. For elements with rotational masses (beam and shell elements), the lumping procedure is element dependent.

Note that the lumping of rotational degrees of freedom is slightly different in implicit and explicit analysis. The rotational masses in

explicit analysis are sometimes scaled up so that they do not affect the element's critical time step.

8.1.4 Damping – central difference method

- Damping can be added directly to the model through Rayleigh damping. Additional indirect damping results from plasticity, friction and rate dependent penalty contact.
- Only mass-proportional Rayleigh damping is available in explicit analysis. Hence, the damping matrix \mathbf{C} in Eq. 8.1-3 is set to:

$$\mathbf{C}_{Rayleigh} = \alpha \mathbf{M}$$

where \mathbf{M} is the total lumped mass matrix. See Ref. KJB, Section 9.3.3, for information about selecting the Rayleigh damping constant α .

8.1.5 Damping – Noh-Bathe method

- For the Noh-Bathe method, stiffness and mass proportional Rayleigh damping can be used:

$$\mathbf{C} = \alpha \mathbf{M} + \beta \mathbf{K}$$

where \mathbf{M} is the total lumped mass matrix and \mathbf{K} is the initial stiffness matrix.

- It should be noted that excessive damping can affect the critical time step for Noh-Bathe method, therefore it is important to properly calculate α, β parameters. One way to do it is to calculate these parameters based on the lowest frequency ω_1 and the highest frequency ω_2 using a damping ratio formulation shown in ref. KJB, page 797. The lowest frequency ω_1 can be calculated by the program in a separate run. The highest frequency ω_2 can be obtained from the critical time, i.e.,

$$\omega_2 = \frac{2}{\Delta t_{cr}}$$

We can then use two equations from which α and β can be calculated:

$$\frac{\alpha + \beta\omega_1^2}{2\omega_1} = \xi_1$$

$$\frac{\alpha + \beta\omega_2^2}{2\omega_2} = \xi_2$$

where ξ_1 and ξ_2 are damping ratios. It has been observed that using in the above equations damping ratios in the order of 1 to 2 percent gives a reasonable amount of damping.

8.2 Stability

ref. KJB
Section 9.4.2

- The stable time step for a single degree of freedom with central difference time integration is

$$\Delta t_{CR} = \frac{T_N}{\pi} = \frac{2}{\omega_N}$$

The stable time step for a finite element assembly is

$$\Delta t \leq \Delta t_{CR} = \frac{T_{N\min}}{\pi} = \frac{2}{\omega_{N\max}} \leq \frac{2}{\omega_{E\max}}$$

where $\omega_{N\max}$ is the highest natural frequency of the system, which is bound by the highest natural frequency of all individual elements in a model $\omega_{E\max}$ (see Ref. KJB, Example 9.13, p. 815).

- For the Noh-Bathe method, the stable time step is:

$$\Delta t \leq \frac{\Delta t_{cr}}{p}$$

- When automatic time step is selected, the time step size is determined for both, CDM and NB methods, according to the following relationship

$$\Delta t = K \times \Delta t_{E \min} = K \times \frac{2}{\omega_{E \max}} \quad (8.2-1)$$

where K is a factor that scales the time step.

- For most element types the critical time step can be expressed in terms of a characteristic length and a material wave speed

$$\Delta t_E = \frac{L}{c} \quad (8.2-2)$$

where the definition of the length L and the wave speed c depend on the element and material type. For all elastic-plastic materials the elastic wave is used. This condition is used in the program instead of actually evaluating the natural frequency in Eq. (8.2-1).

- Note that the critical time step calculated for all elements is only an estimate. For some elements and material combinations it is exact, and for others it is slightly conservative. However, it may not be small enough for excessively distorted elements (3-D solid and shells), and it will therefore need scaling using K factor in Eq (8.2-1).
- The time step also changes with deformation, due to the change in the geometry of the elements and the change in the wave speed through the element (resulting from a change in the material properties).

Truss and cable elements

The critical time step for a 2-node truss element is

$$\Delta t_E = \frac{L}{c}$$

where L is the length of the element, and c is the wave speed through the element

$$c = \sqrt{\frac{E}{\rho}}$$

Beam and iso-beam elements

The critical time step for the beam element is

$$\Delta t_E = \frac{L}{c} / \sqrt{1 + \frac{12I}{AL^2}}$$

where L is the length of the element, A is the element area, I is the largest moment of inertia, and c is the wave speed through the element

$$c = \sqrt{\frac{E}{\rho}}$$

Shell elements

The critical time step for shell elements is

$$\Delta t_E = \frac{L}{c}$$

where L is a characteristic length of the element based on its area and the length of its sides, and c is the planar wave speed through the shell, which for linear isotropic elastic materials is

$$c = \sqrt{\frac{E}{\rho(1-\nu^2)}}$$

The critical time step estimated here is only approximate, and may be too large for excessively distorted shell elements.

3-D solid elements

The critical time step for the 3-D solid elements is

$$\Delta t_E = \frac{L}{c}$$

where L is a characteristic length of the element, based on its volume and the area of its sides, and c is the wave speed through the element. For linear isotropic elastic materials c is given as

$$c = \sqrt{\frac{E(1-\nu)}{\rho(1+\nu)(1-2\nu)}}$$

The critical time step estimated here is only approximate, and may be too large for excessively distorted 3-D solid elements.

Spring elements

The critical time step for a spring element is

$$\Delta t_E = \frac{2}{\omega_N} = 2 \sqrt{\frac{M_1 M_2}{K(M_1 + M_2)}}$$

where M_1 and M_2 are the masses of the two spring nodes and K is its stiffness. Massless springs are not taken into account in the calculation of the stable time step.

Rigid links

These elements are perfectly rigid and therefore do not affect the stability of explicit analysis.

8.3 Time step management

- The stable time step size has a major influence on the total simulation time. Since this time step is determined based on the highest eigenvalue of the smallest element, a single small or excessively distorted element could considerably increase the solution time, even if this element is not relevant to the full model.

Note that the element having the smallest critical time step size is always provided in the output file.

- Ideally, all elements should have similar critical time steps. If the material properties are uniform throughout the model this means that elements should approximately have the same lengths (see Eq. 8.2-2).
- The evaluation of the critical time step for each element takes some computational time. Therefore, it does not need to be performed at every time step. The parameter NCRSTEP in the ANALYSIS DYNAMIC command determines how frequently the critical time step is reevaluated.
- The time step size for explicit analysis can be unduly small for a realistic solution time. Three features are provided to deal with this problem.
- A global mass scaling variable can be applied to all elements in the model (the MASS-SCALE parameter in the ANALYSIS DYNAMIC command). This scale factor is applied to the densities of all elements, except scalar elements where it is applied directly to their mass.
- Mass scaling could also be applied to elements whose automatically calculated initial time step is below a certain value (DTMIN1 parameter in ANALYSIS DYNAMIC). A mass scale factor is then applied to these elements to make their time steps reach DTMIN1. The mass scaling ratio is then held constant for the duration of the analysis. This option is not used when the time step size is user-specified.
- Elements with automatically calculated time step smaller than a specified value (DTMIN2 parameter in ANALYSIS DYNAMIC) can be completely removed from the model. This parameter is useful for extremely small or distorted elements that do not affect the rest of the model. This option is not used when the time step size is user-specified.
- The three parameters explained above (MASS-SCALE, DTMIN1 and DTMIN2) should all be used with great care to

ensure that the accuracy of the analysis is not significantly compromised.

8.4 Tracking solution progress

- Important model parameters such as the memory used by the model, number of degrees of freedom, solution times, minimum stable time step, warning and error messages are all provided in the .out file.
- The program outputs a more summarized time step information to the .msg file.
- For the Noh-Bathe method, the program outputs solution results for $t + \Delta t$ time steps only.

This page intentionally left blank

9. Frequency domain analysis

- The ADINA system includes several capabilities for the characterization of a structural response using frequency domain analysis.
 - ▶ The dynamic response of a structure submitted to a given excitation described by a response spectrum (ground or support motions typical of earthquakes or shocks) can be investigated (see Section 9.1).
 - ▶ A Fourier analysis of the time history of a point in the structure can be performed (see Section 9.2).
 - ▶ Structural response to harmonic (periodic) vibrations or to random vibrations, whether from a base motion or from applied forces, can be analyzed (see Sections 9.3 and 9.4).
 - ▶ The response of an SDOF system connected to the finite element model can also be investigated (see Section 9.5).

Much of the frequency domain analysis is performed by the AUI during post-processing.

9.1 Response spectrum analysis

- Consider the response of a finite element system to an earthquake, with a ground acceleration a_k , into direction k . We assume that all supports of the system are accelerated simultaneously with a_k . The governing equilibrium equations are (see ref. KJB)

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = -\mathbf{M}\mathbf{d}_k a_k(t) \quad (9.1-1)$$

where \mathbf{d}_k is a direction vector with "1" in the portions of the translational degrees of freedom acting into the direction k and "0" in the other portions, and \mathbf{u} is the vector of relative displacements of the finite element system to the ground motion.

If we perform the mode superposition analysis, the governing decoupled equations are

$$\ddot{x}_i + 2\xi_i \omega_i \dot{x}_i + \omega_i^2 x_i = -\Gamma_i^k a_k(t) \quad (9.1-2)$$

where Γ_i^k are the modal participation factors for ground motion loading (see equation (9.1-7) below). The response solution is

$$x_i^k(t) = -\Gamma_i^k \left\{ \frac{1}{\omega_i \sqrt{1 - \xi_i^2}} \int_0^t e^{-\xi_i \omega_i (t-\tau)} \sin \left[\omega_i \sqrt{1 - \xi_i^2} (t - \tau) \right] a_k(\tau) d\tau \right\} \quad (9.1-3)$$

or

$$x_i^k(t) = -\Gamma_i^k D_i^k \quad (9.1-4)$$

If we note that only the lowest modes (up to a cut-off frequency ω_u) contribute to the response, we obtain for the complete response of the system,

$$\mathbf{u} = - \sum_{\omega_i \leq \omega_u} \Gamma_i^k D_i^k \phi_i + \mathbf{u}_r \quad (9.1-5)$$

$$\boldsymbol{\sigma} = - \sum_{\omega_i \leq \omega_u} \Gamma_i^k D_i^k \boldsymbol{\sigma}_i + \boldsymbol{\sigma}_r \quad (9.1-6)$$

where \mathbf{u}_r and $\boldsymbol{\sigma}_r$ are the residual responses due to summing the modal contributions only up to frequency ω_u .

- The modal participation factors are calculated as follows by ADINA:

$$\Gamma_i^k = \phi_i^T \mathbf{M} \mathbf{d}_k \quad (9.1-7)$$

where ϕ_i is the mode shape of mode i and \mathbf{M} is the system mass matrix. Only the active degrees of freedom are included in this calculation. In particular, any mass coupling between active and

deleted degrees of freedom is ignored.

- The modal participation factors have the following properties:

$$1) \quad \sum_{\text{all modes}} \Gamma_i^k \phi_i = \mathbf{d}_k \quad (9.1-8)$$

This formula is used in deriving the residual (static correction) formulas given below (see equation (9.1-29)), and also is used in deriving the floor response spectrum formulas in Section 9.5.

$$2) \quad \sum_{\text{all modes}} \left(\Gamma_i^k \right)^2 = (\mathbf{d}_k)^T \mathbf{M} \mathbf{d}_k \quad (9.1-9)$$

This formula states that the sum of the squares of the modal participation factors for direction k equals the structural mass for direction k , excluding mass associated with fixed degrees of freedom of the structure. Hence the “modal mass” for mode i , direction k , is $\left(\Gamma_i^k \right)^2$.

- In the response spectrum analysis, the actual dynamic response is estimated in the following manner.

Response not including residual terms: Let $S_a(\omega_i, \xi_i)$ be the design response spectrum acceleration, then the maximum accelerations for mode i are

$$(\ddot{\mathbf{u}}_i)_{\max} = \Gamma_i^k S_a(\omega_i, \xi_i) \phi_i \quad (9.1-10)$$

The maximum relative displacements for mode i are

$$(\mathbf{u}_i)_{\max} = \frac{1}{\omega_i^2} \Gamma_i^k S_a(\omega_i, \xi_i) \phi_i \quad (9.1-11)$$

The maximum stresses for mode i are

$$(\boldsymbol{\sigma}_i)_{\max} = \frac{1}{\omega_i^2} \Gamma_i^k S_a(\omega_i, \xi_i) \boldsymbol{\sigma}_i \quad (9.1-12)$$

The maximum element forces and section results for mode i are

$$(\mathbf{F}_i)_{\max} = \frac{1}{\omega_i^2} \Gamma_i^k S_a(\omega_i, \xi_i) \mathbf{F}_i \quad (9.1-13)$$

The maximum reactions for mode i are

$$(\mathbf{R}_i)_{\max} = \frac{1}{\omega_i^2} \Gamma_i^k S_a(\omega_i, \xi_i) \mathbf{R}_i \quad (9.1-14)$$

Let M_i represent any maximum modal response component for mode i . Then to obtain the modal combination response for the selected component considering all modes with circular frequencies less than or equal to ω_u , the following modal combinations are available:

SRSS method (Square Root of the Sum of the Squares):

$$M_{SUM} = \left(\sum_{\omega_i \leq \omega_u} (M_i)^2 \right)^{\frac{1}{2}} \quad (9.1-15)$$

Absolute sum method:

$$M_{SUM} = \sum_{\omega_i \leq \omega_u} |M_i| \quad (9.1-16)$$

Ten percent method:

$$M_{SUM} = \left(\sum_{\omega_i \leq \omega_u} (M_i)^2 + 2 \sum_{\substack{\omega_i \leq \omega_u \\ \omega_j \leq \omega_u}} |M_i M_j| \right)^{\frac{1}{2}} \quad (9.1-17)$$

The second summation is done on all i and j modes having frequencies which are closely spaced, i.e., for which, with $i < j \leq u$,

$$\frac{\omega_j - \omega_i}{\omega_i} \leq 0.1$$

Double sum method, DSC (Discrete Singular Convolution) method:

$$M_{SUM} = \left(\sum_{\omega_r \leq \omega_u} \sum_{\omega_s \leq \omega_u} |M_r M_s| \varepsilon_{rs} \right)^{\frac{1}{2}} \quad (9.1-18)$$

where

$$\varepsilon_{rs} = \left[1 + \left(\frac{\omega'_r - \omega'_s}{\xi'_r \omega_r + \xi'_s \omega_s} \right)^2 \right]^{-1}$$

and

$$\omega'_r = \omega_r \left(1 - \xi_r^2 \right)^{\frac{1}{2}}$$

$$\xi'_r = \xi_r + \frac{2}{t_d \omega_r}$$

and t_d is the duration of the earthquake. In the double sum method, the absolute value signs in (9.1-18) are included; in the DSC method, the absolute value signs in (9.1-18) are omitted.

Algebraic sum method:

$$M_{SUM} = \sum_{\omega_i \leq \omega_u} M_i \quad (9.1-19)$$

CQC (complete quadratic combination) method:

$$M_{SUM} = \left(\sum_{\omega_r \leq \omega_u} \sum_{\omega_s \leq \omega_u} M_r M_s \rho_{rs} \right)^{\frac{1}{2}} \quad (9.1-20)$$

where

$$\rho_{rs} = \frac{8\sqrt{\xi_r \xi_s} (\xi_r + a \xi_s) a^{\frac{3}{2}}}{(1 - a^2)^2 + 4\xi_r \xi_s a (1 + a^2) + 4(\xi_r^2 + \xi_s^2) a^2}$$

and $a = \frac{\omega_s}{\omega_r}$. You have the option to include only those terms

for which $\rho_{rs} \geq \text{CUTOFF}$. $\text{CUTOFF} = 0.0$ implies that all terms are included, $\text{CUTOFF} = 1.0$ implies that only the diagonal terms are included.

- Note: The results obtained with all methods except the algebraic sum method and DSC method depend upon the coordinate system used in modeling the problem. This is because formulas (9.1-15) to (9.1-18) and (9.1-20) are applied component by component and because the sign of each component is lost (either the component is squared or the absolute value of the component is taken).

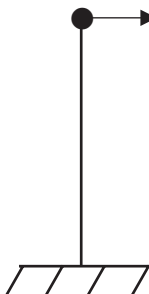
For example, to evaluate the SRSS displacement at a node, (9.1-15) is applied to the maximum x displacement, then to the maximum y displacement and finally to the maximum z displacement. Figure 9.1-1 shows the results for a two-dimensional problem. As shown, the expected result occurs only if one coordinate system direction is aligned with the response direction.

Although the figure illustrates only displacements, the same phenomenon occurs for all results; accelerations, velocities, displacements, reactions, stresses and element forces.

This observation also applies to accelerations, velocities and displacements when skew systems are used at nodes.

Effect of residual (static correction) terms: For the response spectrum method the residual (static correction) terms are calculated using the zero period accelerations S_{ZPA} . Then the following residual terms can be calculated:

Residual accelerations:

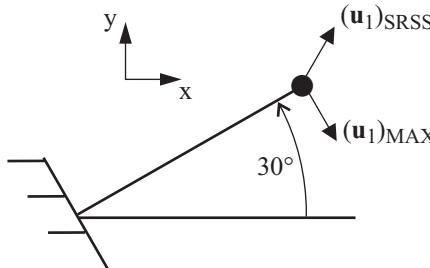
$$\ddot{\mathbf{u}}_r = \left(\mathbf{d}_k - \sum_{\omega_i \leq \omega_u} \Gamma_i^k \boldsymbol{\phi}_i \right) S_{ZPA} \quad (9.1-21)$$


$(\mathbf{u}_1)_{MAX} = (\mathbf{u}_1)_{SRSS}$

$$(\mathbf{u}_1)_{MAX} = \begin{bmatrix} u \\ 0 \end{bmatrix}$$

$$(\mathbf{u}_1)_{SRSS} = \begin{bmatrix} \sqrt{u^2} \\ \sqrt{0^2} \end{bmatrix} = \begin{bmatrix} u \\ 0 \end{bmatrix}$$

a) Model coordinate system aligned with the response direction



$(\mathbf{u}_1)_{SRSS}$

$$(\mathbf{u}_1)_{MAX} = \begin{bmatrix} u \sin 30^\circ \\ -u \cos 30^\circ \end{bmatrix}$$

$$(\mathbf{u}_1)_{SRSS} = \begin{bmatrix} \sqrt{(u \sin 30^\circ)^2} \\ \sqrt{(-u \cos 30^\circ)^2} \end{bmatrix}$$

$$= \begin{bmatrix} u \sin 30^\circ \\ u \cos 30^\circ \end{bmatrix}$$

b) Model coordinate system not aligned with the response direction

Figure 9.1.1: Effects of choice of coordinate system on model combination responses

Residual displacements:

$$\mathbf{u}_r = \mathbf{K}^{-1} \mathbf{M} \left(\mathbf{d}_k - \sum_{\omega_i \leq \omega_u} \Gamma_i^k \boldsymbol{\phi}_i \right) S_{ZPA} \quad (9.1-22)$$

Residual stresses, residual element forces and residual reactions are denoted as $\boldsymbol{\sigma}_r, \mathbf{F}_r, \mathbf{R}_r$ respectively. These residual

quantities will occur when the displacements of the model are equal to the residual displacements \mathbf{u}_r .

- Note that when the residual quantities are graphically displayed separately, then S_{ZPA} is set to unity. When the residual quantities are to be included in a modal combination, however, the expressions shown in Eqs. (9.1-21) and (9.1-22) are, of course, used.
- Let M_r denote a residual response quantity, then the modal combination response quantity, including the residual term, can be calculated as follows:

SRSS addition of residual:

$$M = \left[(M_{SUM})^2 + (M_r)^2 \right]^{\frac{1}{2}} \quad (9.1-23)$$

Absolute addition method including sign: When the SRSS, absolute sum, ten percent, double sum or CQC method is used:

$$M = |(M_{SUM})| + |M_r|$$

When the algebraic sum method is used:

$$M = M_{SUM} + \text{SIGN}(M_r, M_{SUM}) \quad (9.1-24)$$

where

$$\text{SIGN}(M_r, M_{SUM}) = \begin{cases} |M_r| & \text{if } M_{SUM} \geq 0 \\ -|M_r| & \text{if } M_{SUM} < 0 \end{cases}$$

Effect of excitations in multiple directions: In Eqs. (9.1-1) to (9.1-24) we considered the solution of the response due to earthquake excitation in one direction only. To consider excitation in more than one direction, the following spatial combination methods are available:

SRSS addition of spatial components:

$$M_{SPATIAL} = \left[\sum_k (\alpha_k M_k)^2 \right]^{\frac{1}{2}} \quad (9.1-25)$$

Absolute addition of spatial components:

$$M_{SPATIAL} = \sum_k |\alpha_k M_k| \quad (9.1-26)$$

where α_k is a factor for scaling of the component result M_k .

Effect of static load case: Let M_{ST} be a response quantity due to a static load case which has been calculated by ADINA. This static response quantity would represent the condition of the structure at the time when the earthquake occurs and can include, for example, the static response due to gravity load. The final spatial combination can then be obtained as

$$M = M_{ST} + \text{SIGN}(M_{SPATIAL}, M_{ST}) \quad (9.1-27)$$

where

$$\text{SIGN}(M_{SPATIAL}, M_{ST}) = \begin{cases} |M_{SPATIAL}| & \text{if } M_{ST} \geq 0 \\ -|M_{SPATIAL}| & \text{if } M_{ST} < 0 \end{cases} \quad (9.1-28)$$

- Initial calculations for response spectrum analysis must be performed in the ADINA analysis solution; the actual response spectrum analysis being performed during post-processing with the AUI.

Note that any number of steps of linear static solution response calculations can be performed prior to the response spectrum initial calculations in ADINA.

- The initial calculations performed in ADINA are given in the following. Note that these results are stored in the porthole file.
 - The modal participation factors Γ_i^k for each direction k of ground translational excitation and for each eigenmode, using

equation (9.1-7). Note that the number of eigenmodes used in the response spectrum calculation must be less than or equal to the number of eigenmodes available; otherwise, the program issues an error message and stops the execution.

- The modal stress vector \mathbf{s}_i for each specified eigenmode for all elements in which stress output is requested. See Section 6.2.3 for more information.

These stresses do not have a "physical meaning". They are used for calculating the spectral response of the structure.

If the nodal force output option is used in an element, the modal forces corresponding to a displacement vector equal to ϕ_i are instead calculated, printed and stored.

- In addition, initial calculations can be performed for static corrections. If requested, ADINA calculates, prints and stores on the porthole file the following data for each direction k of excitation.

- Residual accelerations $\ddot{\mathbf{u}}_R^k$, where

$$\ddot{\mathbf{u}}_R^k = \mathbf{d}_k - \sum_{i=1}^N \Gamma_i^k \phi_i \quad (9.1-29)$$

- Residual displacements \mathbf{u}_R^k for the total structural system, where

$$\mathbf{u}_R^k = {}^{t_0}\mathbf{K}^{-1} \mathbf{M} \ddot{\mathbf{u}}_R^k \quad (9.1-30)$$

The definitions for the variables on the right side of the equation are given above and ${}^{t_0}\mathbf{K}$ is the stiffness matrix at the start time of the solution.

- Residual forces \mathbf{F}_R^k for the total structural system, where

$$\mathbf{F}_R^k = \mathbf{M} \ddot{\mathbf{u}}_R^k \quad (9.1-31)$$

- ▶ Residual stresses \mathbf{s}_R^k for all elements which specify stress output, where

$$\mathbf{s}_R^k = {}^{t_0}\mathbf{C} {}^{t_0}\mathbf{B} \mathbf{u}_R^k \quad (9.1-32)$$

and ${}^{t_0}\mathbf{C}$, ${}^{t_0}\mathbf{B}$ are the element stress-strain matrix and strain-displacement matrix, respectively at the start time of the solution.

If the nodal force output option is used in an element, the residual forces corresponding to a displacement vector equal to \mathbf{u}_R^k are instead calculated, printed and stored.

- ▶ Residual reactions, consistent with the residual displacements.
- During post-processing the AUI performs the response spectrum analysis. You specify the following information during post-processing, to obtain the response for a single ground motion direction:
 - ▶ Loading response spectra (in terms of displacements, velocities or accelerations) for various damping ratios
 - ▶ Ground motion direction
 - ▶ Damping ratios of the modes
 - ▶ Type of modal combination
 - ▶ Type of combination of residual terms

In the AUI, all of the information needed for the calculation for one ground motion direction is entered into a response of type response-spectrum.

To perform spatial combination calculations for more than one ground motion direction, define a response of type response-combination and include the response spectrum responses for the individual ground motion directions in the definition of the response-combination response.

To include the effects of a static load case, define a response of type load-step for the static load case, then define a response of type response-combination and include the response spectrum responses and the static load case response in the definition of the response-combination response.

See Section 13.1.4 for information about responses in the AUI.

- Additional information about response spectrum analysis can be found in the following references:

ref. R.W. Clough and J. Penzien, *Dynamics of Structures*, McGraw Hill, 1974.

ref. U.S. Nuclear Regulatory Guide 1.92, *Combining Modal Responses and Spatial Components in Seismic Response Analysis*, Revision 1, February 1976.

9.2 Fourier analysis

- Given a time history $f(t)$, defined from t_0 to t_1 , the Fourier series corresponding to $f(t)$ is

$$f(t) = a_0 + \sum_{n=1}^{\infty} (a_n \cos n\omega_1 t + b_n \sin n\omega_1 t)$$

where

$$\omega_1 = \frac{2\pi}{t_p}, \quad t_p = t_1 - t_0$$

$$a_0 = \frac{1}{t_p} \int_{t_0}^{t_1} f(t) dt$$

$$a_n = \frac{2}{t_p} \int_{t_0}^{t_1} f(t) \cos n\omega_1 t dt, \quad b_n = \frac{2}{t_p} \int_{t_0}^{t_1} f(t) \sin n\omega_1 t dt$$

Since

$$a_n \cos n\omega_1 t + b_n \sin n\omega_1 t = c_n \cos(n\omega_1 t - \phi_n)$$

where

$$c_n = \sqrt{a_n^2 + b_n^2}, \quad \phi_n = \tan^{-1} \frac{b_n}{a_n}$$

$f(t)$ may be written as

$$f(t) = c_0 + \sum_{n=1}^{\infty} c_n \cos(n\omega_1 t - \phi_n)$$

- Conceptually, the function $f(t)$ is considered to contain the frequencies $0, \omega_1, 2\omega_1, \dots$. The constants c_n and ϕ_n are the amplitude and phase angle of that portion of $f(t)$ which oscillates at frequency $n\omega_1$.
- A simple relationship (Parseval's identity) exists between the Fourier coefficients and the mean square amplitude of $f(t)$:

$$\frac{1}{T} \int_{t_0}^{t_1} f(t)^2 dt = c_0^2 + \frac{1}{2} \sum_{n=1}^{\infty} c_n^2$$

- The power spectral density of $f(t)$ for frequency $f_n = \frac{n\omega_1}{2\pi}$ may be defined as

$$S_f(f_0) = \frac{c_0^2}{\Delta f}$$

$$S_f(f_n) = \frac{c_n^2}{2\Delta f}, \quad n = 1, 2, \dots$$

where $\Delta f = \frac{\omega_1}{2\pi}$ is the interval between two successive frequencies

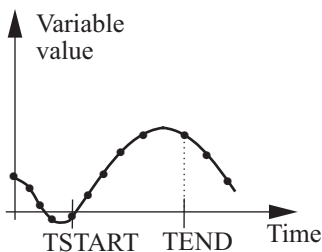
(in cycles/unit time). The mean square value of $f(t)$ may be expressed in terms of the power spectral density using

$$\frac{1}{t_p} \int_{t_0}^{t_1} f^2 dt = \sum_{n=1}^{\infty} S_f(f_n) \Delta f$$

It is seen that the mean square value of $f(t)$ is related to the area under the power spectral density curve.

- The AUI permits $f(t)$ to be defined as the value of any variable over any time interval contained within the solution time. The time step need not be constant in the finite element analysis. The AUI permits an additional time interval t_{gap} during which $f(t)$ is zero to be added to the variable response. These concepts are shown in Fig. 9.2-1.

Time history of variable:



Time history used in Fourier analysis:

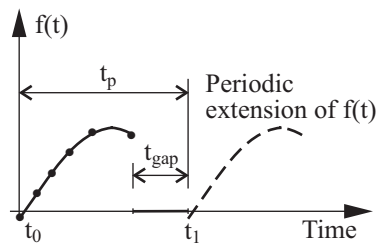


Figure 9.2-1: Fourier analysis

- The Fourier series is truncated by choosing a highest frequency f_{max} . This frequency is chosen based on the time step to avoid aliasing errors:

$$f_{max} \leq \frac{0.5}{\Delta t|_{\max}}$$

Hence, $N = \text{int}(t_p f_{max})$ gives the upper limit of the Fourier series summation

$$f(t) = c_0 + \sum_{n=1}^N c_n \cos(n\omega_1 t - \phi_n)$$

- The Fourier coefficient integrals are evaluated using the trapezoidal rule for numerical integration. These integrals are typically integrated with less accuracy for higher frequencies. Because a fast Fourier transform algorithm is not used, the computer time required to form the Fourier coefficients is proportional to N^2 .

9.3 Harmonic vibration analysis

- It is frequently of interest to determine the steady-state structural response to given harmonic and random loads. This structural response is conveniently expressed as the superposition of modal responses. This section summarizes the theory of harmonic vibration analysis and discusses those features of the AUI that perform harmonic vibration analysis.
- Consider the finite element system

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \sum_{k=1}^N \mathbf{F}_k(t) \quad (9.3-1)$$

where N load cases are being considered. If we perform a mode superposition analysis, the governing decoupled equations are

$$\ddot{x}_i + 2\xi_i \omega_i \dot{x}_i + \omega_i^2 x_i = \sum_{k=1}^N \Gamma_i^k b_k(t) \quad (9.3-2)$$

where Γ_i^k is the modal participation factor for load case k and mode i .

These equations apply for the following cases:

Case 1: Structure is grounded and forces $\mathbf{F}_k(t)$ are applied.

Here k is the load case number:

$$\mathbf{F}_k(t) = \mathbf{F}_{k0} b_k(t) \quad (9.3-3)$$

where $b_k(t)$ is a multiplier for load case k . The modal participation factor is $\Gamma_i^k = \boldsymbol{\phi}_i^T \mathbf{F}_{k0}$. When \mathbf{F}_{k0} includes mass-proportional loads, the mass-proportional load calculation does not include the mass coupling between active and deleted degrees of freedom.

Case 2: A single ground motion is applied in direction k :

$$\mathbf{F}_k(t) = -\mathbf{M} \mathbf{d}_k b_k(t) \quad (9.3-4)$$

where $b_k(t)$ is the ground acceleration in direction k . The modal participation factor is $\Gamma_i^k = -\boldsymbol{\phi}_i^T \mathbf{M} \mathbf{d}_k$. Only the active degrees of freedom are included in this calculation. In particular, any mass coupling between active and deleted degrees of freedom is ignored.

Case 2 is seen to be a special case of case 1, except that in case 1 the displacements are interpreted as absolute displacements and in case 2 the displacements are interpreted as relative displacements.

Each load case is seen to have a spatial variation and a time variation. The time variation appears within the function $b_k(t)$. In harmonic analysis, we consider functions $b_k(t)$ of the form

$$b_k(t) = b_{k0} \sin(\omega_k t - \alpha_k) \quad (9.3-5)$$

where b_{k0} is a function of the forcing frequency ω_k .

- We also note here that the transient structure response is ignored. In most cases, only the mean-square value of a response quantity can be predicted. However, in harmonic vibration analysis with all loads acting with the same loading frequency, the maximum amplitude and phase of a response quantity can be predicted as well.

Harmonic vibration analysis with all loads acting with the same loading frequency

- Consider the time variation

$$b_k(t) = b_{k0} \sin(\omega_k t - \alpha_k) \quad (9.3-6)$$

for load case k , in which each b_{k0} is a function of the forcing frequency, and we write ω instead of ω_k because the forcing frequency ω is assumed to be the same for each load case k . In this case, the steady-state modal response is

$$x_i(t) = \sum_{k=1}^N (A_{ki} \sin(\omega t - \alpha_k) - B_{ki} \cos(\omega t - \alpha_k)) \quad (9.3-7)$$

where

$$A_{ki} = \frac{b_{k0} \Gamma_i^k}{\omega_i^2} \frac{1 - \left(\frac{\omega}{\omega_i}\right)^2}{\left(1 - \left(\frac{\omega}{\omega_i}\right)^2\right)^2 + \left(2 \frac{\omega}{\omega_i} \xi_i\right)^2} \quad (9.3-8)$$

$$B_{ki} = \frac{b_{k0} \Gamma_i^k}{\omega_i^2} \frac{2 \frac{\omega}{\omega_i} \xi_i}{\left(1 - \left(\frac{\omega}{\omega_i}\right)^2\right)^2 + \left(2 \frac{\omega}{\omega_i} \xi_i\right)^2} \quad (9.3-9)$$

and the steady-state solution is then

$$\mathbf{u}(t) = \sum_{i=1}^n \boldsymbol{\phi}^{(i)} \sum_{k=1}^N x_{ki}(t) \quad (9.3-10)$$

where n is the number of eigenvectors considered in the mode superposition. Equation (9.3-10) is of course only an approximation to the steady-state solution response when n is less than the total number of degrees of freedom in the model. However the approximation may be adequate when enough eigenvectors are considered and when harmonic loads of low enough frequency are considered.

If we consider a single degree of freedom L , then

$$\begin{aligned} u_L(t) &= \sum_{i=1}^n \Phi_L^{(i)} \sum_{k=1}^N x_{ki}(t) \\ &= \left(\sum_{i=1}^n \Phi_L^{(i)} \sum_{k=1}^N C_{ki} \right) \sin \omega t - \left(\sum_{i=1}^n \Phi_L^{(i)} \sum_{k=1}^N D_{ki} \right) \cos \omega t \end{aligned} \quad (9.3-11)$$

where

$$\begin{aligned} C_{ki} &= A_{ki} \cos \alpha_k - B_{ki} \sin \alpha_k \\ D_{ki} &= A_{ki} \sin \alpha_k + B_{ki} \cos \alpha_k \end{aligned}$$

which can be rewritten as

$$u_L(t) = u_{L0} \sin(\omega t - \theta_L) \quad (9.3-12)$$

where

$$u_{L0} = \sqrt{\left(\sum_{i=1}^n \Phi_L^{(i)} \sum_{k=1}^N C_{ki} \right)^2 + \left(\sum_{i=1}^n \Phi_L^{(i)} \sum_{k=1}^N D_{ki} \right)^2} \quad (9.3-13)$$

$$\theta_L = \tan^{-1} \left(\frac{\sum_{i=1}^n \Phi_L^{(i)} \sum_{k=1}^N D_{ki}}{\sum_{i=1}^n \Phi_L^{(i)} \sum_{k=1}^N C_{ki}} \right) \quad (9.3-14)$$

u_{L0} and θ_L are seen to be the maximum amplitude and phase angle, respectively.

- The quasi-static amplitude of the displacements for forcing frequency ω is determined by assuming that the load $b_k(t)$ is applied with a very low forcing frequency. One way to determine the quasi-static amplitude for forcing frequency ω is to set ω to zero within equations (9.3-8) and (9.3-9) (but using b_{k0} corresponding to forcing frequency ω) and then evaluate equation (9.3-13) as described above. This method is approximate because the static structural response may not be well modeled in a modal superposition analysis; in other words, the contribution of the remaining modes to the static response may not be negligible.

The ratio of the dynamic maximum amplitude to the quasi-static maximum amplitude is sometimes referred to as the dynamic magnification factor or the dynamic load factor.

- Equations (9.3-11) to (9.3-14) also apply to any quantity that is proportional to the displacements, such as element forces, stresses and reaction forces.
- Considering the velocity and acceleration, we see from equation (9.3-12) that

$$\dot{u}_L(t) = \omega u_{L0} \sin\left(\omega t - \left(\theta_L - \frac{\pi}{2}\right)\right) \quad (9.3-15)$$

and

$$\ddot{u}_L(t) = \omega^2 u_{L0} \sin(\omega t - (\theta_L - \pi)) \quad (9.3-16)$$

from which the maximum amplitude and phase are easily seen.

- For more insight into the above equations, we consider only one load case ($N=1$) and draw (9.3-6) and (9.3-12) in the complex plane (Fig. 9.3-1). Notice that the actual response is the imaginary component of the vectors.

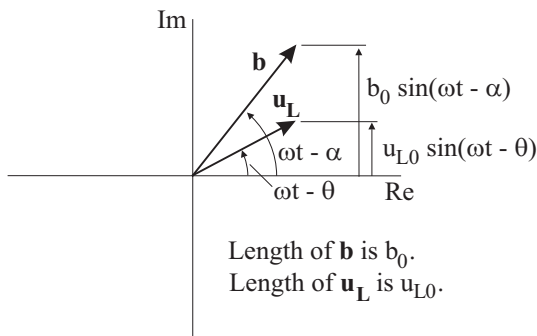


Figure 9.3-1: Force and displacement representations in the complex plane

To obtain the “in-phase” or “real” part of the response, we set $\omega t - \alpha_k$ to 90° . This orients the force vector to be entirely imaginary (Fig 9.3-2). The in-phase part of the response is then obtained from (9.3-12).

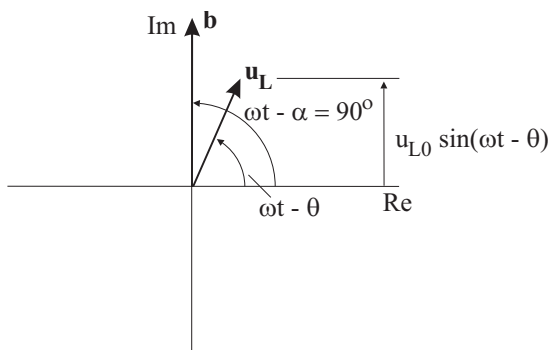


Figure 9.3-2: Obtaining the in-phase part of the displacements

To obtain the “out-of-phase” or “imaginary” part of the response, we set $\omega t - \alpha_k$ to 0° . This orients the force vector to be entirely real (Fig 9.3-3). The in-phase part of the response is then obtained from (9.3-12).

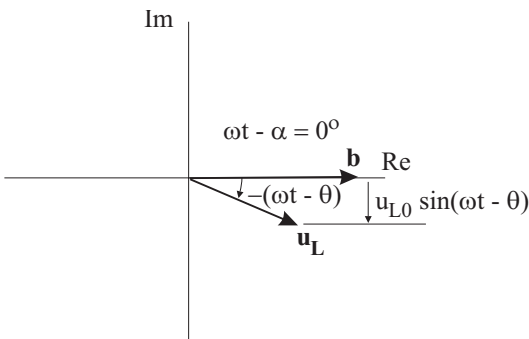


Figure 9.3-3: Obtaining the out-of-phase part of the displacements

The reason why we use 90° to obtain the real part of the response is because equations (9.3-6) and (9.3-12) are written in terms of the sine function, not the cosine function.

- Additional information about harmonic vibration analysis can be found in the following references:

ref. R.W. Clough and J. Penzien, *Dynamics of Structures*, McGraw-Hill, 1974.

ref. W.C. Hurty and M.F. Rubinstein, *Dynamics of Structures*, Prentice Hall, 1964.

- Concerning the usage of the harmonic vibration analysis feature, ADINA computes the natural frequencies, mode shapes (including modal stresses and forces) and modal participation factors. The type of modal participation factor can either be applied load or ground motion. When the type of modal participation factor is applied load, ADINA computes the load vector for each time step in the analysis as if the analysis was a linear static analysis. However ADINA does not solve for the displacements corresponding to the load vector, rather ADINA computes the modal participation factor.
- During post-processing the AUI performs the harmonic vibration analysis. You specify the following information during post-processing:

- ▶ Sweep spectrum giving the dependence of b_{k0} on the forcing frequency
- ▶ Load step number (for applied loads) or ground motion direction (for ground motion loading)
- ▶ Damping ratios of the modes
- ▶ Type of result (maximum amplitude, phase angle, RMS amplitude or signed amplitude for a given value of ωt)

Then you can obtain either the entire structural response for a single forcing frequency or the response at a point for a range of forcing frequencies.

- In order to obtain the in-phase or out-of-phase part of the response, select signed amplitude for the type of result and set ωt to 90° (to obtain the in-phase part of the response) or to 0° (to obtain the out-of-phase part of the response). (We assume here that the loading phase angle α_k is equal to 0.)
- The harmonic vibration analysis information is entered into a response of type harmonic. See Section 13.1.4 for information about responses in the AUI.

9.4 Random vibration analysis

- For general information about the theory of random vibration analysis, see reference:

ref. W.C. Hurty and M.F. Rubinstein, *Dynamics of Structures*, Prentice-Hall, 1964.

- In the ADINA system, the only type of result that can be predicted in random vibrations is the mean-square response, defined as

$$\overline{p^2} = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T p^2(t) dt \quad (9.4-1)$$

for the function $p(t)$. The mean-square response can also be written as an integral in the frequency domain as

$$\overline{p^2} = \lim_{T \rightarrow \infty} \frac{1}{2\pi} \int_0^T \frac{|\tilde{p}(\omega)|^2}{T} d\omega \quad (9.4-2)$$

in which

$$\tilde{p}(\omega) = \int_{-\infty}^{\infty} p(t) \exp(-i\omega t) dt \quad (9.4-3)$$

is the Fourier transform of $p(t)$ and

$$|\tilde{p}(\omega)|^2 = \tilde{p}(\omega) \tilde{p}^*(\omega) \quad (9.4-4)$$

where $\tilde{p}^*(\omega)$ is the complex conjugate value of $\tilde{p}(\omega)$. The mean-square response in the frequency domain can also be written in terms of the power spectral density function

$$\hat{y}(\omega) = \frac{|\tilde{y}(\omega)|^2}{T} \quad (9.4-5)$$

as follows:

$$\overline{y^2(t)} = \frac{1}{2\pi} \int_0^{\infty} \hat{y}(\omega) d\omega = \int_0^{\infty} \hat{y}(f) df \quad (9.4-6)$$

where f is the frequency in cycles/(unit time). Using Fourier transform theory, the Fourier transform of the modal response can be written in terms of the Fourier transform of the loads:

$$\tilde{x}_i(\omega) = \sum_{k=1}^N \Gamma_i^k \tilde{H}_i(\omega) \tilde{b}_k(\omega) \quad (9.4-7)$$

where

$$\tilde{H}_i(\omega) = \frac{1}{(\omega_i^2 - \omega^2) + i(2\xi_i \omega_i \omega)} \quad (9.4-8)$$

Therefore, for component u_L ,

$$\tilde{u}_L(\omega) = \sum_{i=1}^n \phi_L^{(i)} \sum_{k=1}^N \Gamma_i^k \tilde{H}_i(\omega) \tilde{b}_k(\omega) \quad (9.4-9)$$

and therefore

$$\begin{aligned} |\tilde{u}_L(\omega)|^2 = & \\ & \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^N \sum_{l=1}^N \phi_L^{(i)} \phi_L^{(j)} \Gamma_i^k \Gamma_j^l \tilde{H}_i(\omega) \tilde{H}_j^*(\omega) \tilde{b}_k(\omega) \tilde{b}_l^*(\omega) \end{aligned} \quad (9.4-10)$$

The following approximations are made to simplify (9.4-10):

Approximation 1)

$$\tilde{b}_k(\omega) \tilde{b}_l^*(\omega) \approx |\tilde{b}_k(\omega)|^2 \delta_{kl} \quad (9.4-11)$$

Equation (9.4-11) is equivalent to neglecting the cross-spectral densities of the loading combinations.

Approximation 2)

$$\tilde{H}_i(\omega) \tilde{H}_j^*(\omega) \approx |\tilde{H}_i(\omega)| |\tilde{H}_j(\omega)| \quad (9.4-12)$$

Equation (9.4-12) is equivalent to ignoring the phase relationships between the response functions.

For more information about approximation 2, see Hurty & Rubenstein, specifically the discussion after equation (11.75) in

that reference.

Using eqs (9.4-11) to (9.4-12), equation (9.4-10) reduces to

$$\left| \tilde{u}_L(\omega) \right|^2 = \sum_{k=1}^N \sum_{i=1}^n \sum_{j=1}^n \left(\phi_L^{(i)} \right) \left(\phi_L^{(j)} \right) \left(\Gamma_i^k \right) \left(\Gamma_j^k \right) \left| \tilde{H}_i(\omega) \right| \left| \tilde{H}_j(\omega) \right| \left| \tilde{b}_k(\omega) \right|^2 \quad (9.4-13)$$

Using the definition of the power-spectral-density, we write

$$\hat{u}_L(\omega) = \sum_{k=1}^N \sum_{i=1}^n \sum_{j=1}^n \left(\phi_L^{(i)} \right) \left(\phi_L^{(j)} \right) \left(\Gamma_i^k \right) \left(\Gamma_j^k \right) \left| \tilde{H}_i(\omega) \right| \left| \tilde{H}_j(\omega) \right| \hat{b}_k(\omega) \quad (9.4-14)$$

and therefore

$$\overline{u_L^2} = \sum_{k=1}^N \sum_{i=1}^n \sum_{j=1}^n \left(\phi_L^{(i)} \right) \left(\phi_L^{(j)} \right) \left(\Gamma_i^k \right) \left(\Gamma_j^k \right) \int_0^\infty \left| \tilde{H}_i(f) \right| \left| \tilde{H}_j(f) \right| \hat{b}_k(f) df \quad (9.4-15)$$

In addition to the two approximations (9.4-11) and (9.4-12), ADINA also allows the user to make an additional approximation in equations (9.4-14) and (9.4-15):

Approximation 3)

$$\left| \tilde{H}_i(\omega) \right| \left| \tilde{H}_j(\omega) \right| = 0 \quad \text{for} \quad \frac{\min(\omega_i, \omega_j)}{\max(\omega_i, \omega_j)} < \text{CUTOFF} \quad (9.4-16)$$

When CUTOFF = 0, no approximation is made and when CUTOFF = 1, eq. (9.4-16) reduces to

$$\left| \tilde{H}_i(\omega) \right| \left| \tilde{H}_j(\omega) \right| = \left| \tilde{H}_i(\omega) \right|^2 \delta_{ij}$$

The default is CUTOFF=0. Approximation 3 with CUTOFF=1 is discussed before equation (11.79) in Hurty & Rubenstein.

In (9.4-15) and (9.4-16), the input power-spectral-density $\hat{b}_k(f)$ satisfies

$$\overline{b_k^2(t)} = \frac{1}{2\pi} \int_0^\infty \hat{b}_k(\omega) d\omega = \int_0^\infty \hat{b}_k(f) df \quad (9.4-17)$$

where $\overline{b_k^2(t)}$ is the mean-square value of the applied load (compare (9.4-6) and (9.4-17)). Different authors use different conventions, for example

$$\begin{aligned} \overline{b_k^2(t)} &= \int_0^\infty \hat{b}_k(\omega) d\omega, \\ \overline{b_k^2(t)} &= \int_{-\infty}^\infty \hat{b}_k(f) df, \\ \overline{b_k^2(t)} &= \int_{-\infty}^\infty \hat{b}_k(\omega) d\omega \end{aligned}$$

are all possible definitions of the input power-spectral-density. However, ADINA accepts only the convention given in (9.4-17).

Equations (9.4-14) and (9.4-15) are used when the response quantity is proportional to the displacements. When the response quantity is proportional to the velocities or the accelerations, then the transfer function given in equation (9.4-8) has to be modified:

$$\text{Velocity : } \tilde{H}_i(\omega) = \frac{i\omega}{(\omega_i^2 - \omega^2) + i(2\xi_i\omega_i\omega)}$$

$$\text{Acceleration : } \tilde{H}_i(\omega) = \frac{-\omega^2}{(\omega_i^2 - \omega^2) + i(2\xi_i\omega_i\omega)}$$

Interpretation of mean-square responses: The mean-square value of a response quantity can be used to predict the probability that the actual response will exceed any given value. This is done by assuming that the response has a Gaussian distribution and has

zero mean value. Then it may be shown that the standard deviation is well approximated by the root-mean-square (RMS) value of the response quantity:

$$\sigma = \sqrt{u_L^2} \quad (9.4-18)$$

Hence we can write

$$P_r(|u_L| > \lambda \sigma) = \operatorname{erfc}\left(\frac{\lambda}{\sqrt{2}}\right) \quad (9.4-19)$$

where the value of the erfc function can be found in standard tables:

$$P_r(|u_L| > \sigma) = 0.3173$$

$$P_r(|u_L| > 2\sigma) = 0.0455$$

$$P_r(|u_L| > 3\sigma) = 0.0027$$

- Concerning the usage of the random vibration analysis feature, ADINA computes the same quantities as in harmonic vibration analysis (see above).
- During post-processing the AUI performs the random vibration analysis. You specify the following information during post-processing:
 - ▶ Power-spectral-density $\hat{b}_k(f)$. Note that $\hat{b}_k(f)$ must satisfy (9.4-17). If the input power-spectral-density is obtained using another convention, you must convert it so that (9.4-17) is satisfied.
 - ▶ Load step number (for applied loads) or ground motion direction (for ground motion loading)
 - ▶ Damping ratios of the modes

Then you can obtain either the RMS response or the power-spectral-density of the response at a point.

- The random vibration analysis information is entered into a response of type random. See Section 13.1.4 for information about responses in the AUI.

9.5 SDOF system response

- It is frequently of interest to determine the maximum response of single degree of freedom (SDOF) systems to SDOF support displacement excitation as a function of the SDOF system natural frequency.
- The support is a node in the finite element mesh (called the connection point) and the SDOF systems are considered to be attached to the support. The SDOF systems are assumed to be much lighter than the structure, so that the SDOF system response does not affect the structural response.
- This type of analysis is called floor response spectrum analysis. The maximum SDOF system response can be computed when the structure time history is given, or when the response spectrum corresponding to structural ground motion is given.
- As an example of this type of analysis, consider a machine within a building that is much lighter than the building. Because the machine is much lighter than the building, it is assumed that the machine response does not affect the building response, and therefore the machine response calculations can be done as a postprocessing step once the building response (without the machine) is known.

The response of the building to an earthquake can be computed either using a time history analysis (when the earthquake ground motion is given as a function of time) or using a response spectrum analysis (when the earthquake ground motion is given as a response spectrum). The response of the machine can also be computed in each case (Figs. 9.5-1 and 9.5-2) by idealizing the machine as a SDOF system attached to a point in the building.

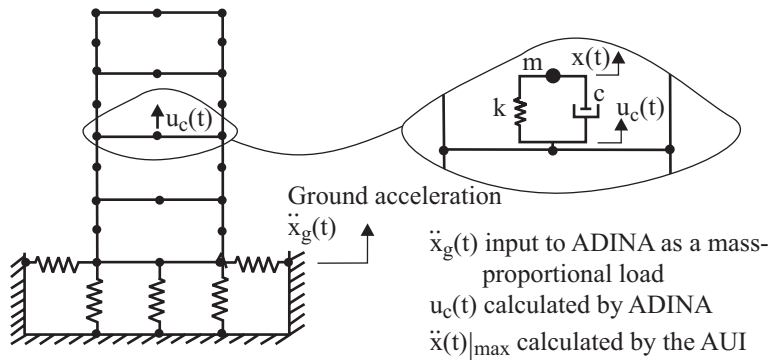
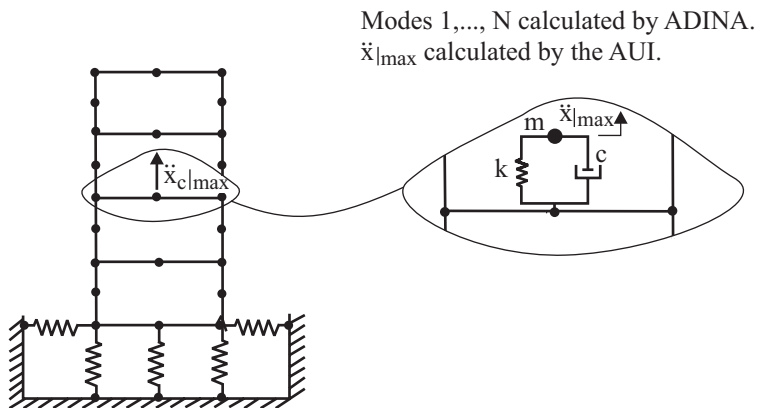


Figure 9.5-1: Floor response spectrum analysis from time history data



Ground acceleration given by response spectrum:

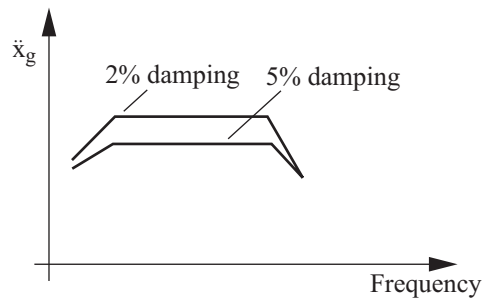


Figure 9.5-2: Floor response spectrum analysis from response spectrum data

- Considering many possible machines, each with a different natural frequency and damping ratio, it is useful to create floor response spectrum curves (Fig. 9.5-3). Each floor response spectrum curve shows, for a given machine damping ratio, the maximum machine response as a function of the machine natural frequency. One possible use of the floor response spectrum curves is in the selection of a machine mount that will minimize the machine's motion in the event of an earthquake, for example, by selecting the mount stiffness so that the machine has the natural frequency with the least response.

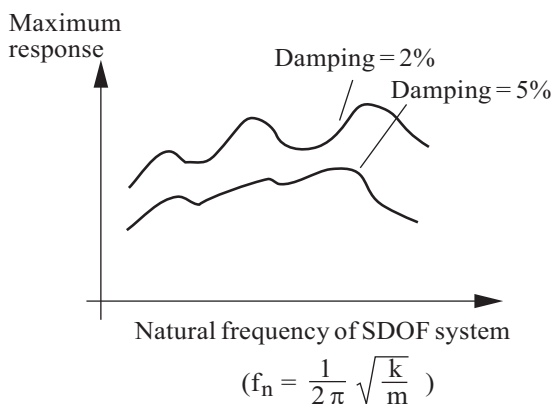


Figure 9.5-3: Floor response spectrum curves

Structural loading given by one or more time histories

- The governing equation of motion for the SDOF system is

$$m\ddot{u} + c\dot{u} + ku = ku_c + c\dot{u}_c - m\ddot{x}_g \quad (9.5-1)$$

where u is the relative displacement of the SDOF system to ground motions applied as mass-proportional loads, u_c is the relative displacement of the connection point to ground motions applied as mass-proportional loads and \ddot{x}_g are ground accelerations applied as mass-proportional loads. Equation (9.5-1) is written in terms of relative displacements because the nodal displacements, velocities and accelerations output by ADINA are relative to the ground

motion when ground motions are applied as mass-proportional loads.

Equation (9.5-1) is numerically integrated using the Newmark method with $\delta = 1/2$, $\alpha = 1/4$ from rest initial conditions and the maximum response is recorded. The maximum response can be the maximum values of $|u|, |\dot{u}|, |\ddot{u}|, |u - u_c|, |\dot{u} - \dot{u}_c|, |\ddot{u} - \ddot{u}_c|, |x|, |\dot{x}|,$

or $|\ddot{x}|$. The connection point displacement, velocity and acceleration are taken directly from the ADINA output, so no numerical integration is required to evaluate these quantities. When the absolute displacement or velocity of the SDOF system are required, the AUI numerically integrates the ground accelerations to obtain ground velocities and displacements using the trapezoidal rule.

The damping ratio used for the SDOF systems can be constant or frequency-dependent.

- The AUI can combine the maximum responses from several time histories using the floor response spectrum from time history command. In this case, the AUI computes the maximum response separately for each time history and combines the results using either the SRSS method or the absolute sum method (see Section 9.1).

This feature is most useful when you have several separate time history analyses of a structure as separate ADINA runs, each time history analysis giving the structural response to loading acting in a different direction, and you want to compute the maximum response corresponding to the loads acting simultaneously. First, load the ADINA porthole files into a single database, using the "append" feature to load all but the first porthole file. Then compute the response corresponding to combined loads, in which you define response-ranges for each time history, then specify each response-range in a separate data input line. (Note: appending the porthole files shifts the time intervals associated with each time history so that the time intervals do not overlap.)

- The spacing of SDOF system frequencies can be linear, logarithmic or the recommended USNRC or ASME frequency spacings, see references:

ref. *1992 ASME Boiler and Pressure Vessel Code, Rules for*

*Construction of Nuclear Power Plant Components,
Division 1, Appendix N.*

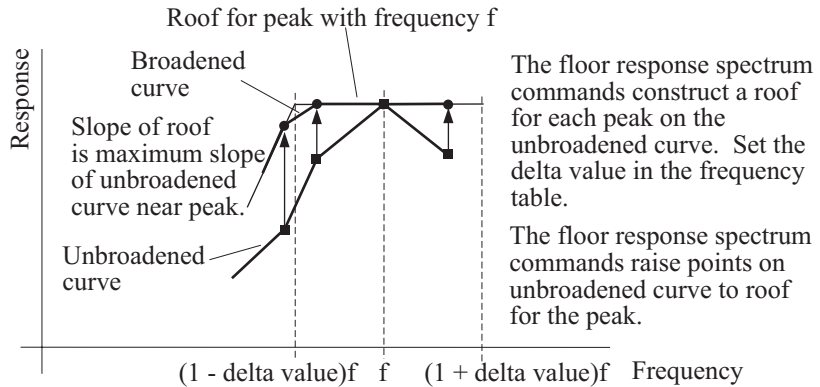
- ref. U.S. Nuclear Regulatory Guide 1.122, *Development of Floor Design Response Spectra for Seismic Design of Floor-Supported Equipment or Components*, rev. 1, Feb. 1978.

In addition, the SDOF system response is evaluated for each frequency listed in a frequency table. This feature is useful for evaluating the response at structural natural frequencies.

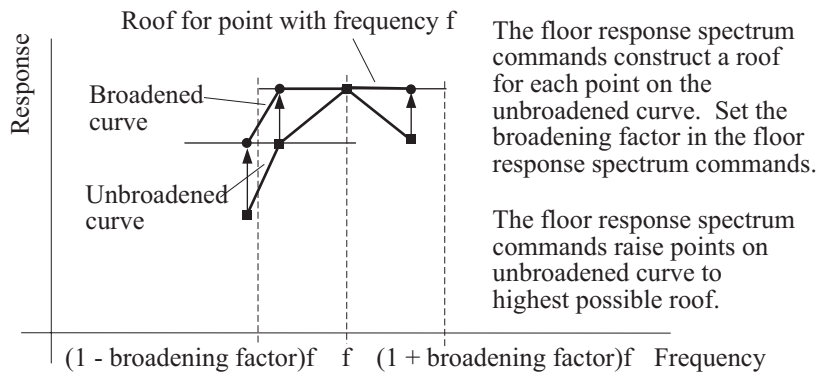
- After the AUI constructs a response curve, it can broaden peaks and smooth the curve. There are two peak broadening algorithms that you can use: peak broadening=yes and peak broadening=all. These are shown in Figure 9.5-4.

During smoothing, response values are raised (never lowered) from their unsmoothed values until the curve segments are as close to straight as possible. By default, it is set to 5 % of the maximum response value in the curve.

Peak broadening can be used without smoothing, or smoothing can be used without peak broadening, but these features are most effective when used together.



a) Broadening algorithm used when peak broadening = yes



b) Broadening algorithm used when peak broadening = all

Figure 9.5-4: Floor response spectrum broadening algorithms

Structural loading specified by response spectra

- Frequently the structural loading is one or more ground motions specified as response spectra. In this case it is not possible to compute the structural response using a time history analysis. Instead the structural response is estimated using the response spectrum method (see Section 9.1).

The technique, which is presented in the following reference,

- ref. K.A. Peters, D. Schmitz and U. Wagner, "Determination of Floor Response Spectra on the Basis of the Response Spectrum Method," *Nuclear Engineering and Design*, Vol. 44, pp. 255-262, 1977.

can be summarized as follows:

Start with the eigenvalues, eigenvectors, modal participation factors and modal damping ratios of the structure. These are termed

$$\omega_i, \phi_i, \Gamma_i^k, \xi_i$$

for modes $i = 1, \dots, N$ and ground motion direction k . The eigenvectors are of length n , where n is the number of degrees of freedom of the structure. For example, the eigenvalues, eigenvectors and modal participation ratios are calculated by ADINA in response spectrum analysis and the modal damping ratios are specified in the AUI (ADINA-PLOT).

The addition of a SDOF system to the structure contributes an extra eigenvalue, eigenvector, modal participation factor and modal damping factor to the mathematical model. These are termed

$$\omega_i^*, \phi_i^*, (\Gamma_i^k)^*, \xi_i^*$$

where the * is used to denote the structure plus SDOF system mathematical model. The addition of the SDOF system also modifies the structure eigenvalues, eigenvectors, modal participation factors and modal damping factors for modes $i = 1, \dots, N$:

$$\begin{aligned} \omega_i &\rightarrow \omega_i^*, & \phi_i &\rightarrow \phi_i^*, \\ \Gamma_i^k &\rightarrow (\Gamma_i^k)^*, & \xi_i &\rightarrow \xi_i^* \end{aligned}$$

The eigenvectors are now of length $n + 1$, with the additional entry in each eigenvector denoted ϕ_{i0}^* .

Consider a SDOF system that vibrates in direction ℓ . In order to estimate the maximum absolute acceleration of the SDOF system

due to ground motion in direction k , we use the response spectrum formula

$$\left(\ddot{x}_0^k\right)_{\max} = \text{modal-combination of } \left(\ddot{x}_{0i}^k\right)_{\max}, \quad i = 0, \dots, N \quad (9.5-2)$$

where

$$\left(\ddot{x}_{0i}^k\right)_{\max} = \left(\Gamma_i^k\right)^* \phi_{i0}^* S_a\left(\omega_i^*, \xi_i^*\right) \quad (9.5-3)$$

is the maximum acceleration due to mode i and $S_a\left(\omega_i^*, \xi_i^*\right)$ is the response spectrum acceleration corresponding to natural frequency ω_i^* and damping ratio ξ_i^* . Any of the modal combination methods listed in Section 9.1 can be used for the modal combination formula. For example, if the SRSS method of modal combination is used,

$$\left(\ddot{x}_0^k\right)_{\max} = \left(\sum_{i=0}^N \left[\left(\Gamma_i^k\right)^* \phi_{i0}^* S_a\left(\omega_i^*, \xi_i^*\right) \right]^2 \right)^{\frac{1}{2}} \quad (9.5-4)$$

We see from eq. (9.5-3) that we need $\left(\Gamma_i^k\right)^*, \phi_{i0}^*, i = 1, \dots, N$ and $\left(\Gamma_0^k\right)^* \phi_{00}^*$ to apply the response spectrum method to the structure plus SDOF system mathematical model. From the last reference above, if the SDOF system is very light compared with the structure,

$$\Phi_i^* \approx \begin{bmatrix} \Phi_i \\ \overline{\phi_{i0}^*} \end{bmatrix} \quad (9.5-5a)$$

$$\omega_i^* \approx \omega_i, \quad \left(\Gamma_i^k\right)^* \approx \Gamma_i^k, \quad \xi_i^* \approx \xi_i, \quad i = 1, \dots, N \quad (9.5-5b,c,d)$$

Also,

$$\phi_{i0}^* \approx \frac{1}{1 - \left(\frac{\omega_i}{\omega_0}\right)^2} \phi_{ia} \quad (9.5-6)$$

where a denotes the degree of freedom corresponding to the node to which the SDOF system is connected and direction ℓ . (We note that the fraction in this equation is similar to the transfer function for a SDOF system with natural frequency ω_0 and zero damping subjected to harmonic support motion). Finally,

$$\left(\Gamma_0^k\right)^* \phi_{00}^* = \delta_{k\ell} - \sum_{i=1}^n \left(\Gamma_i^k\right)^* \phi_{i0}^* \quad (9.5-7)$$

where $\delta_{k\ell} = 1$ for $k = \ell$ and $\delta_{k\ell} = 0$ otherwise. Equation (9.5-7) follows from equation (9.1-8).

Since we normally have fewer than n eigenvectors, we make the approximation

$$\left(\Gamma_0^k\right)^* \phi_{00}^* \approx \delta_{k\ell} - \sum_{i=1}^N \left(\Gamma_i^k\right)^* \phi_{i0}^* \quad (9.5-8)$$

by neglecting

$$\sum_{i=N+1}^n \left(\Gamma_i^k\right)^* \phi_{i0}^* = \sum_{i=N+1}^n \Gamma_i^k \frac{1}{1 - \left(\frac{\omega_i}{\omega_0}\right)^2} \phi_{ia} \quad (9.5-9)$$

From the right-hand-side of equation (9.5-9), we see that the neglected term is small provided that the natural frequency of the SDOF system is much smaller than the natural frequencies of the neglected modes.

Introducing equations (9.5-5), (9.5-6) and (9.5-7) into equation (9.5-3) gives the response spectrum approximation to the maximum absolute acceleration of the SDOF system. We term this

approximation the undamped transfer function, no resonance correction method because equation (9.5-6) corresponds to an undamped transfer function and there is no check for the SDOF system natural frequency being close to a structural natural frequency.

When the SDOF system natural frequency is close to a structural natural frequency, the denominator of equation (9.5-6) tends to zero. In this case, we use the following equations

$$\begin{aligned} \left(\ddot{x}_0^k\right)_{\max} &= \text{modal-combination of } \left(\dot{x}_{0i}^k\right)_{\max} \text{ and } \left(\dot{x}_{0j}^k\right)_{\max}, \\ i &= 1, \dots, N, \quad i \neq j \end{aligned} \quad (9.5-10)$$

where

$$\left(\dot{x}_{0j}^k\right)_{\max} = \frac{\left(\Gamma_j^k\right)^* \phi_{ja}^* S_a\left(\omega_j^*, \xi_j^*\right)}{2\sqrt{\xi_0\left(\xi_0 + \xi_j\right)}} \quad (9.5-11)$$

is an estimate of the maximum acceleration of the SDOF system due to the motion of that mode j with natural frequency closest to the natural frequency of the SDOF system. It is recommended that the results using both equations (9.5-2) and (9.5-10) be calculated and the smaller result taken; we term this method the undamped transfer function, resonance correction method.

We have found that equation (9.5-10) can be smaller than equation (9.5-2) even for SDOF system natural frequencies not close to any structural natural frequency, for example when the structural motion of mode j does not affect the SDOF system motion. When this happens, the response predictions are not accurate. Therefore we introduce a third method, the damped transfer function method, which is identical to the undamped transfer function, no resonance correction method, except that equation (9.5-6) is replaced by

$$\phi_{i0}^* \approx \pm \frac{1 + \left(2\xi_{eff} \frac{\omega_i}{\omega_0} \right)^2}{\sqrt{\left(1 - \left(\frac{\omega_i}{\omega_0} \right)^2 \right)^2 + \left(2\xi_{eff} \frac{\omega_i}{\omega_0} \right)^2}} \phi_{ia} \quad (9.5-12)$$

where + is used if $\omega_i \leq \omega_0$ and - is used otherwise. The term ξ_{eff} in this equation is evaluated using one of the following choices:

$$\xi_{eff} = \xi_0 \quad (9.5-12a)$$

$$\xi_{eff} = \sqrt{\xi_0 (\xi_0 + \xi_j)} \quad (9.5-12b)$$

$$\xi_{eff} = \sqrt{\xi_0^2 \pm (\xi_0 - \xi_j)^2}, \text{ + if } \xi_0 \leq \xi_j, \text{ - if } \xi_0 > \xi_j \quad (9.5-12c)$$

Equation (9.5-12) corresponds to the transfer function for a damped SDOF system subjected to support motion. The damped transfer function method avoids numerical difficulties when the SDOF system natural frequency equals a structural natural frequency and gives results close to the undamped transfer function, no resonance correction method otherwise, provided that the SDOF damping ratio is small (which is usually the case).

- The maximum relative velocity and maximum relative displacement corresponding to ground motions in direction k can be estimated using similar techniques by substituting the spectrum velocity S_v or spectrum displacement S_d for the spectrum acceleration S_a in the above formulas. Note that, although the calculated acceleration is an absolute acceleration, the velocities and displacements are relative to the ground motion.
- Once the responses are known for a single load case, the response for a combination of load cases can be computed as follows:

$$(\ddot{x}_0)_{\max} = \text{spatial-combination of } (\ddot{x}_0^k)_{\max}, \quad k = 1, \dots, N \quad (9.5-13)$$

For example, when the SRSS method is used to combine three load cases, each load case corresponding to ground motions in one of the three coordinate directions,

$$(\ddot{x}_0)_{\max} = \left[\sum_{k=1}^3 (\ddot{x}_0^k)^2 \right]^{\frac{1}{2}} \quad (9.5-14)$$

- The options available for selecting SDOF system frequencies, damping values, broadening peaks and smoothing curves are exactly the same as when calculating floor response spectrum curves using time history data, see above.
- Note on the use of the above theory with the CQC method of modal combinations. When the above theory is used, along with equation (9.5-12), and along with the CQC method, the resonances tend to be underestimated. This is because, in the CQC method, the magnitude of the predicted resonance depends on the cancellation of the quantities M_r , where M_r is the maximum modal response component for mode r . To be specific, as two modes r and s approach each other, the contribution of these modes to the CQC sum approaches $(M_r + M_s)^2$. When equation (9.5-12) is used, then M_r and M_s remain finite, with nearly equal magnitudes, and opposite signs, hence the CQC sum tends to zero. But when equation (9.5-6) is used, then M_r and M_s grow unbounded, with nearly equal magnitudes, and opposite signs, so that the CQC sum remains large.

This page intentionally left blank

10. Fracture mechanics

10.1 Overview

- Linear and nonlinear fracture mechanics analysis can be performed with the ADINA system.

The following capabilities are included:

- Computation of the J-integral in 2-D finite element models, using the line contour method. See Section 10.9 for more details.
- Computation of the J-integral in 2-D finite element models, using the virtual crack extension method with nodal virtual shifts. See Section 10.10 for more details.
- The analysis of actual crack propagation in 2-D finite element models with symmetric geometry and loading. See Section 10.11 for more details.
- Computation of the J-integral in 3-D finite element models, using the virtual crack extension method with nodal virtual shifts (NVS method). See Section 10.14 for more details.
- Computation of the J-integral and stress intensity factors K_I , K_{II} , K_{III} in 3-D finite element models, using the virtual crack extension method with station virtual shifts (SVS method). See Section 10.15 for more details.
- The fracture mechanics computations can be performed for the following element types: **2-D solid** and **3-D solid**. The material models that can be used are: **isotropic-elastic**, **orthotropic-elastic**, **thermo-isotropic**, **thermo-orthotropic**, **plastic-orthotropic**, **plastic-bilinear**, **plastic-multilinear**, **thermo-plastic**, **creep**, **plastic-creep**, **multilinear-plastic-creep**, **creep-variable**, **plastic-creep-variable**, **multilinear-plastic-creep-variable**, **plastic-cyclic**, **user-supplied**. Any of the kinematic formulations that are allowed for the chosen materials can be used: **small displacement/small strain**, **large displacement/small strain**, and **large displacement/large strain**. The only exception is that the

ULJ formulation cannot be used for large strain analysis of plastic materials.

Table 10.1-1 gives some of the basic features for ADINA fracture mechanics:

Table 10.1-1: Basic features for ADINA fracture mechanics

	Line contour	NVS method	SVS method
2-D model, stationary crack	✓	✓	
2-D model, propagating crack	✓	✓	
3-D model		✓	✓
More than one crack in the same analysis			✓
Calculation of the J-integral	✓	✓	✓
Calculation of stress intensity factors			✓ (note 1)
Mapped mesh near crack front	✓ (notes 2, 3)	✓ (notes 2, 3)	✓ (note 3)
Free-form mesh near crack front			✓
Blunt crack front (coincident nodes on the crack front)	✓	✓	

Note 1: Only when the elements that are in the SVS domain surrounding the crack are all linear elastic isotropic without thermal effects, and when the analysis is static.

Note 2: See Section 10.14 for a detailed description of the meshing required for the NVS method in 3-D analysis.

Note 3: The CRACK-M features for meshing 3-D cracks can be used both with the NVS and SVS methods.

- Additional information about the theory used in ADINA can be found in the following references:

- ref. T.L. Anderson, *Fracture Mechanics, Fundamentals and Applications*, 3rd edition, CRC Press, 2005.
- ref. S.N. Atluri, "Path-independent integrals in finite elasticity and inelasticity, with body forces, inertia, and arbitrary crack-face conditions," *Engng. Fracture Mech.*, Vol. 16, pp. 341-364, 1982.
- ref. H.G. deLorenzi, 'On the energy release rate and the J-integral for 3-D crack configurations', *Int. J. Fract.*, Vol. 19, pp. 183-193, 1982.
- ref. O. Guillermin, "Some Developments in Computational Methods for Nonlinear Fracture Mechanics," *Proceedings IKOSS FEM '91*, Baden-Baden, Germany, 1991.
- ref. T.K. Hellen, "On the Method of Virtual Crack Extensions," *Int. J. Num. Meth. Engng.*, Vol. 9, pp. 187-207, 1975.
- ref. J.K. Knowles and E. Sternberg, "On a Class of Conservation Laws in Linearized and Finite Elastostatics," *Archiv. Rat. Mech. Anal.*, Vol. 44, pp. 187-211, 1972.
- ref. D.M. Parks, "The Virtual Crack Extension Method for Nonlinear Material Behavior," *Comp. Meth. Appl. Mech. Engng.*, Vol. 12, pp. 353-364, 1977.
- ref. J.R. Rice, "A Path Independent Integral and the Approximate Analysis of Strain Concentration by Notches and Cracks," *J. Appl. Mech.*, Vol. 35, pp. 379-386, 1968.
- ref. C.F. Shih, B. Moran, T. Nakamura, "Energy release rate along a three-dimensional crack front in a thermally stressed body", *Int. J. Fract.*, Vol. 30, pp. 79-102, 1986.

- ref. C.F. Shih, R. J. Asaro, "Elastic-Plastic Analysis of Cracks on Bimaterial Interfaces: Part I — Small Scale Yielding", *J. Appl. Mech.*, Vol. 55, pp. 299-316, 1988.
- ref. M.C. Walters, G. H. Paulino, R.H. Dodds, "Interaction Integral Procedures for 3-D Curved Cracks Including Surface Traction" *Engr. Fract. Mech.*, Vol. 72, pp. 1635-1663, 2005.

10.2 The energy release rate

The energy release rate is defined as

$$G = -\frac{d\Pi}{dA} \quad (10.2-1)$$

where Π is the total potential energy and A is the crack area. Π is defined by

$$\Pi = U - \sum Ru \quad (10.2-2)$$

where U is the internal energy (strain energy) and $\sum Ru$ is the sum of the work done by each applied load R moving through its corresponding displacement u (with the load R held constant as the displacement increases from 0 to u). Note that A is the area of one side of the crack, e.g, if the crack has length a and thickness b , the crack area is $A = ab$.

- The operation $\frac{d\Pi}{dA}$ is interpreted as follows:

$$\frac{d\Pi}{dA} = \lim_{\Delta A \rightarrow 0} \frac{\Pi|_{A+\Delta A} - \Pi|_A}{\Delta A} \quad (10.2-3)$$

that is, the difference of the total potential energy for two slightly different crack areas divided by the change in crack area.

- Although, in theory, the energy release rate could be calculated using two finite element analyses and (10.2-3), the energy release rate is more efficiently calculated using the line contour or virtual crack extension methods, as described below.

10.3 Stress intensity factors

Consider a straight crack within a linear isotropic material under static conditions without thermal effects or body forces. The displacement and stress fields close to the crack tip under plane stress or plane strain conditions can be written in terms of three fundamental fields, corresponding to the three fundamental modes of deformation I, II, III. These three fundamental modes of deformation are shown in Fig. 10.3-1.

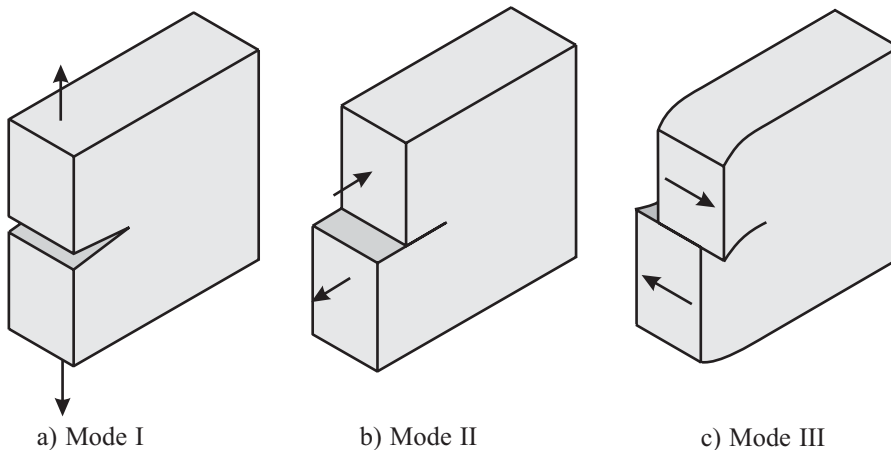


Figure 10.3-1: Fundamental modes of deformation near the crack front

Mode I:

$$u_1^I = \frac{1}{2\mu} \sqrt{\frac{r}{2\pi}} \left[\cos \frac{\theta}{2} \left(\kappa - 1 + 2 \sin^2 \frac{\theta}{2} \right) \right] \quad (10.3-1a)$$

$$u_2^I = \frac{1}{2\mu} \sqrt{\frac{r}{2\pi}} \left[\sin \frac{\theta}{2} \left(\kappa + 1 - 2 \cos^2 \frac{\theta}{2} \right) \right] \quad (10.3-1b)$$

$$u_3^I = 0 \quad (10.3-1c)$$

$$\tau_{11}^I = \frac{1}{\sqrt{2\pi r}} \left[\cos \frac{\theta}{2} \left(1 - \sin \frac{\theta}{2} \sin \frac{3\theta}{2} \right) \right] \quad (10.3-1d)$$

$$\tau_{22}^I = \frac{1}{\sqrt{2\pi r}} \left[\cos \frac{\theta}{2} \left(1 + \sin \frac{\theta}{2} \sin \frac{3\theta}{2} \right) \right] \quad (10.3-1e)$$

$$\tau_{12}^I = \frac{1}{\sqrt{2\pi r}} \left[\cos \frac{\theta}{2} \sin \frac{\theta}{2} \cos \frac{3\theta}{2} \right] \quad (10.3-1f)$$

$$\tau_{13}^I = 0 \quad (10.3-1g)$$

$$\tau_{23}^I = 0 \quad (10.3-1h)$$

Mode II:

$$u_1^{II} = \frac{1}{2\mu} \sqrt{\frac{r}{2\pi}} \left[\sin \frac{\theta}{2} \left(\kappa + 1 + 2 \cos^2 \frac{\theta}{2} \right) \right] \quad (10.3-2a)$$

$$u_2^{II} = \frac{1}{2\mu} \sqrt{\frac{r}{2\pi}} \left[-\cos \frac{\theta}{2} \left(\kappa - 1 - 2 \sin^2 \frac{\theta}{2} \right) \right] \quad (10.3-2b)$$

$$u_3^{II} = 0 \quad (10.3-2c)$$

$$\tau_{11}^{II} = \frac{1}{\sqrt{2\pi r}} \left[-\sin \frac{\theta}{2} \left(2 + \cos \frac{\theta}{2} \cos \frac{3\theta}{2} \right) \right] \quad (10.3-2d)$$

$$\tau_{22}^{II} = \frac{1}{\sqrt{2\pi r}} \left[\sin \frac{\theta}{2} \cos \frac{\theta}{2} \cos \frac{3\theta}{2} \right] \quad (10.3-2e)$$

$$\tau_{12}^{II} = \frac{1}{\sqrt{2\pi r}} \left[\cos \frac{\theta}{2} \left(1 - \sin \frac{\theta}{2} \sin \frac{3\theta}{2} \right) \right] \quad (10.3-2f)$$

$$\tau_{13}^{II} = 0 \quad (10.3-2g)$$

$$\tau_{23}^{II} = 0 \quad (10.3-2h)$$

Mode III:

$$u_1^{III} = 0 \quad (10.3-3a)$$

$$u_2^{III} = 0 \quad (10.3-3b)$$

$$u_3^{III} = \frac{1}{\mu} \sqrt{\frac{2r}{\pi}} \sin \frac{\theta}{2} \quad (10.3-3c)$$

$$\tau_{11}^{III} = 0 \quad (10.3-3d)$$

$$\tau_{22}^{III} = 0 \quad (10.3-3e)$$

$$\tau_{12}^{III} = 0 \quad (10.3-3f)$$

$$\tau_{13}^{III} = -\frac{1}{\sqrt{2\pi r}} \sin \frac{\theta}{2} \quad (10.3-3g)$$

$$\tau_{23}^{III} = \frac{1}{\sqrt{2\pi r}} \cos \frac{\theta}{2} \quad (10.3-3h)$$

In the above, $\mu = \frac{E}{2(1+\nu)}$ is the shear modulus,

$$\kappa = \begin{cases} 3-4\nu & \text{plane strain} \\ \frac{3-\nu}{1+\nu} & \text{plane stress} \end{cases} \quad (10.3-4)$$

and, for future reference,

$$E' = \begin{cases} \frac{E}{1-\nu^2} & \text{plane strain} \\ E & \text{plane stress} \end{cases} \quad (10.3-5)$$

In all cases,

$$\tau_{33} = \begin{cases} \nu(\tau_{11} + \tau_{22}) & \text{plane strain} \\ 0 & \text{plane stress} \end{cases} \quad (10.3-6)$$

These formulas refer to a polar coordinate system with origin at the crack, as shown in Fig. 10.3-2.

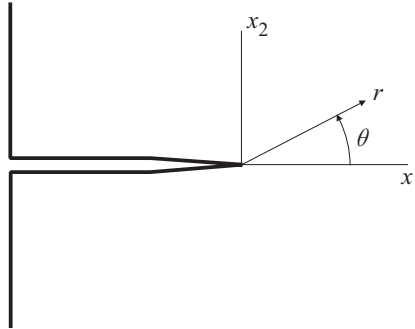


Figure 10.3-2: Geometry and coordinate system for stress intensity factor fields

These formulas are well-known and can be found in the book by Anderson, see the list of references given above.

By superposition, the displacement and stress fields close to the crack tip can be written in terms of these fundamental fields and in terms of three multipliers K_I , K_{II} , K_{III} :

$$u_i = K_I u_i^I + K_{II} u_i^{II} + K_{III} u_i^{III} + u_i^R \quad (10.3-7)$$

$$\tau_{ij} = K_I \tau_{ij}^I + K_{II} \tau_{ij}^{II} + K_{III} \tau_{ij}^{III} + \tau_{ij}^R \quad (10.3-8)$$

K_I , K_{II} , K_{III} are referred to as the stress intensity factors. Terms u_i^R , τ_{ij}^R are referred to as "residual" fields, and are expected to be much smaller than the other terms, near the crack tip. The residual terms are added here in anticipation of the developments in Section 10.6.2.

The dimensions of the multipliers K_I , K_{II} , K_{III} are $[\text{stress}]\sqrt{[\text{length}]}$, so that the units in the SI system are $\text{MPa}\sqrt{\text{m}}$.

A positive value of K_I is associated with an opening of the crack, and a negative value of K_I is associated with closing of the crack. Therefore it is expected that K_I is positive. However, the other multipliers K_{II} , K_{III} can either be positive or negative.

10.4 Calculation of the J-integral using the line contour integration method

- The line contour integration method is used in two-dimensional analysis to calculate a contour independent parameter (J-integral) characterizing the severity of the displacement, stress and strain fields at the tip of a crack. The J-integral is given by (see Fig. 10.4-1)

$$J = \int_{\Gamma} \left(W dx_2 - \tau_{ij} n_j \frac{\partial u_i}{\partial x_1} ds \right) \quad (10.4-1)$$

where Γ = line contour enclosing the crack tip, τ_{ij} = components of the stress tensor, u_i = components of the displacement vector, n_j = components of the unit vector normal to Γ , ds = length increment along Γ , W = strain energy density. It is assumed that the crack is oriented along the x_1 direction.

The detailed calculation of the strain energy density is discussed in Section 11.7.

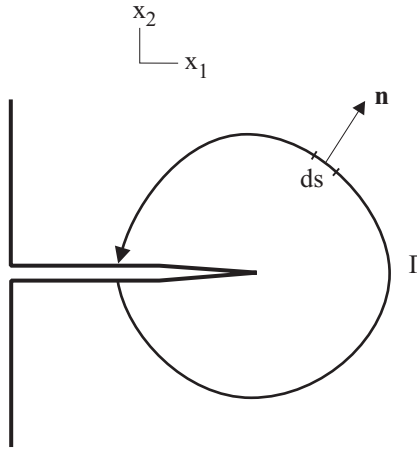


Figure 10.4-1: Arbitrary contour around a crack tip for J-integral calculation

- Note that initial strains are not included in the computation of the J-integral.
- For linear elastic materials, the J-integral is related to the stress intensity factors K_I , K_{II} and K_{III} described in Section 10.3 by

$$J = \frac{K_I^2}{E'} + \frac{K_{II}^2}{E'} + \frac{K_{III}^2}{2\mu} \quad (10.4-2)$$

- The J-integral as defined above is equivalent to the energy release rate under the following conditions:
 - ▶ Isothermal analysis
 - ▶ No body forces and inertia forces; no pressure on the crack faces
 - ▶ Monotonic and proportional loading; no unloading
 - ▶ Material homogeneous in the x_1 direction

- The J-integral concept can be extended to large displacement and/or large strain conditions. The J-integral becomes

$$J = \int_{\Gamma} \left({}^0W \, {}^0dx_2 - {}^0T_{ji} \, {}^0n_j \frac{\partial u_i}{\partial {}^0x_1} {}^0ds \right) \quad (10.4-3)$$

in which the left superscript / subscript 0 denotes “in the original configuration”, or “referred to the original configuration”. ${}^0T_{ij}$ are the components of the 1st Piola-Kirchhoff stress tensor, defined as

$${}^0T_{ij} = \left(\det {}^t\mathbf{X} \right) {}^0X_{ik} \tau_{kj} \quad (10.4-4)$$

- Under large displacement and/or large strain conditions, J is path-independent only when the net forces acting on the contour are zero, i.e.

$$\int_{\Gamma} {}^0T_{ji} \, {}^0n_j \, {}^0ds = 0 \quad (10.4-5)$$

Usually this is the case for contours used in fracture mechanics.

10.5 Calculation of the J-integral using the virtual crack extension method

- In the virtual crack extension method, the J-integral is calculated using the expression

$$G = \frac{1}{\Delta A_c} \int_V \left(\tau_{ij} \frac{\partial u_j}{\partial x_k} - W \delta_{ik} \right) \frac{\partial \Delta X_k}{\partial x_i} dV \quad (10.5-1)$$

where V = volume of the cracked body, ΔX_k = components of the virtual crack shift vector, ΔA_c = increase in crack area corresponding to ΔX_k , δ_{ij} = Kronecker delta, and W = strain energy density. The calculation of W is discussed in Section 11.7. In (10.5-1), we have not yet included several corrections described

in more detail below. If there are initial strains, these initial strains are used in the computation of the stresses and strain energy density.

- We use the symbol G to denote the J-integral when the J-integral is calculated using a volume integral, as in (10.5-1). Other authors use the symbol J to denote the J-integral regardless of how the J-integral is calculated.

- The calculation of G using (10.5-1) relies on the definition of a virtual shift vector field $\Delta\mathbf{X}$, which is defined throughout the volume. The basic concept of the virtual shift vector field is shown in Figs 10.5-1 and 10.5-2. On the crack front, the virtual shift vector field specifies the virtual shift of the crack. Within a tube that surrounds the crack front, the virtual shift vector field specifies the shift of material particles close to the crack front. Outside the tube, the virtual shift vector field is zero, and material particles outside of the tube are not shifted.

The virtual shift vector field $\Delta\mathbf{X}$ is assumed to vary smoothly throughout the tube, so that this field can be differentiated as required in equation (10.5-1).

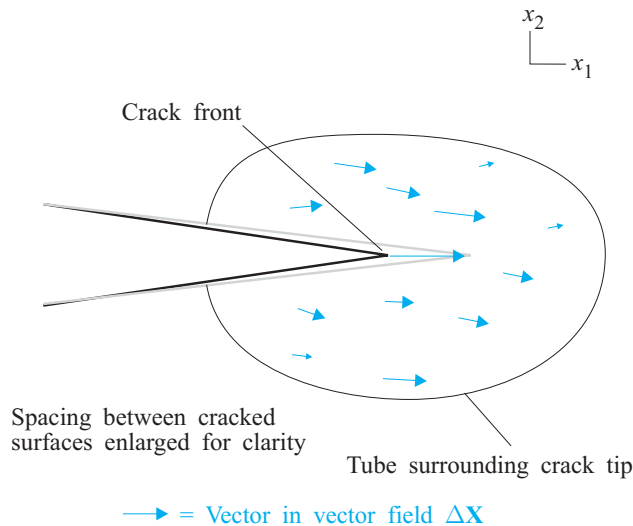


Figure 10.5-1: The virtual crack extension method in two-dimensional analysis

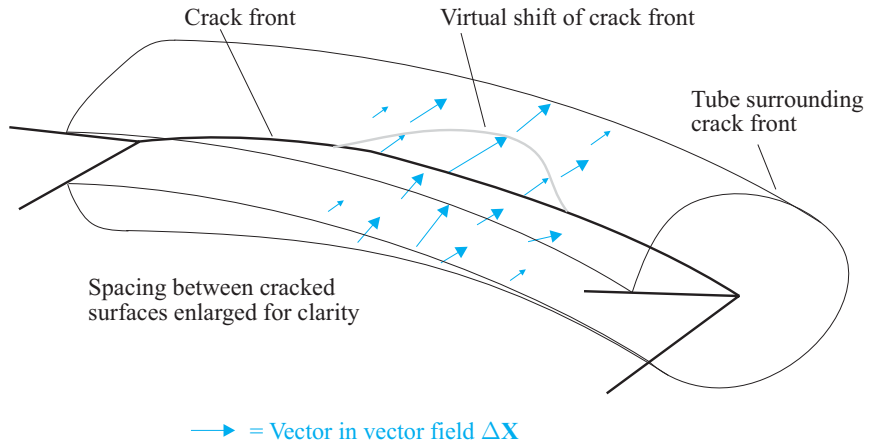


Figure 10.5-2: The virtual crack extension method in three-dimensional analysis

In three-dimensional analysis, it is convenient to introduce an orthonormal coordinate system aligned with the crack front, as shown in Fig. 10.5-3). Direction **a** is normal to the crack front, in the plane of self-similar crack propagation, direction **b** defines the plane of self-similar crack propagation and direction **c** is tangent to the crack front.

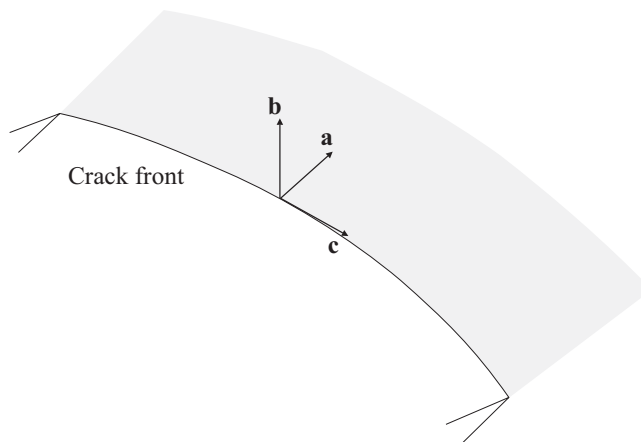


Figure 10.5-3: Crack front coordinate system

The virtual shift vector field $\Delta \mathbf{X}$ causes the crack area to increase according to the formula

$$\Delta A_c = \int \mathbf{a} \cdot \Delta \mathbf{X} ds \quad (10.5-2)$$

where ds is the differential length along the crack front (Fig. 10.5-4) and the integration is performed along the crack front line.

Notice that it is not necessary for $\Delta \mathbf{X}$ to be perpendicular to the crack front, however $\Delta \mathbf{X}$ must lie in the plane defined by direction \mathbf{b} .

For a 2-D crack, this formula simplifies to become

$$\Delta A_c = t \|\Delta \mathbf{X}\| = t \sqrt{\Delta X_1^2 + \Delta X_2^2} \quad (10.5-3)$$

where t is the thickness at the crack tip ($t = x_1$ for an axisymmetric model). $\Delta \mathbf{X}$ must lie in the line of self-similar crack advance.

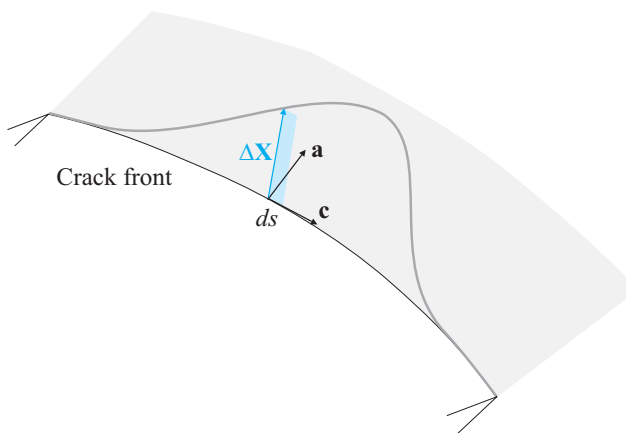


Figure 10.5-4: Calculation of virtual crack area increase

Because the integral in (10.5-1) is proportional to the magnitude of $\Delta \mathbf{X}$, and because ΔA_c is also proportional to the magnitude of $\Delta \mathbf{X}$, the value of G is unaffected by the magnitude of $\Delta \mathbf{X}$.

- The theoretical value of G is not affected by the size or shape of the tube, or by the values of the virtual shift field $\Delta \mathbf{X}$ away from the crack front line. Thus G , like the J-integral, is said to be path-independent.

Hoop stress correction: In axisymmetric analysis with 2-D elements, the following correction must be added to G :

$$G_{hoop} = \frac{1}{\Delta A_c} \int_V \left(\tau_{xx} \frac{\partial u_x}{\partial x} - W \right) \Delta X_2 dV \quad (10.5-4)$$

where ΔX_2 is the virtual crack extension component in the y direction.

Thermal correction: For the analysis of structures with temperature dependent material properties, the effect of thermal loads must be taken into account in the calculation of the J-integral. The following correction should be added to G :

$$G_{th} = \frac{1}{\Delta A_c} \int_V \tau_{ij} \frac{\partial e_{ij}^{th}}{\partial \theta} \frac{\partial \theta}{\partial x_k} \Delta X_k dV \quad (10.5-5)$$

in which e_{ij}^{th} are the components of the thermal strain tensor and θ is the temperature. Note that, for those materials in which the material constants are temperature-dependent, the effect of material constants changing with temperature is not included. (However, the effect of thermal coefficient changing with temperature is included.)

Pressure correction: When there are tractions acting on the crack faces, the following correction should be added to G :

$$G_{pr} = -\frac{1}{\Delta A_c} \int_S t_i \frac{\partial u_i}{\partial x_j} \Delta X_j dS \quad (10.5-6)$$

where t_i = components of the applied surface traction vector and S = surface of the cracked body. For example when pressures are applied, $t_i = -pn_i$, in which p is the pressure and n_i is the outwards surface normal.

Note that tractions from contact conditions and tractions resulting

from fluid loadings in FSI analysis are not included in equation (10.5-6).

Dynamic correction: In dynamic analysis, the effect of inertia forces is included in the following correction, which should be added to G :

$$G_{dyn} = \frac{1}{\Delta A_c} \int_V \rho \ddot{u}_i \frac{\partial u_i}{\partial x_j} \Delta X_j dV \quad (10.5-7)$$

in which ρ is the mass density and \ddot{u}_i are the components of the acceleration vector.

Large displacements / large strains: The above expressions are modified for large displacements / large strains by referring all quantities to the original configuration, and by using the transpose of the 1st Piola-Kirchhoff stresses instead of the Cauchy stresses. Note that the virtual shift vector field $\Delta \mathbf{X}$ is always defined in the original configuration.

- G plus all of the above corrections is equivalent to the energy release rate $-\frac{d\Pi}{dA}$ under the following conditions:
 - ▶ No body forces.
 - ▶ Monotonic, proportional loading in inelastic analysis
- Since the virtual shift vector function is arbitrary, the virtual shift vector field $\Delta \mathbf{X}$ can be specified in different ways. There are two fundamental approaches that can be taken:
 - ▶ Virtual shift vectors specified at the nodes, then these vectors interpolated within the elements using the element shape functions. This technique is widely used. In ADINA, this technique is termed the NVS (nodal virtual shift) algorithm, and is described in Section 10.10 for 2-D cracks and in Section 10.14 for 3-D cracks.

Because the virtual shift vectors are specified at the nodes, it

is necessary to employ structured meshes at the crack front.

- Virtual shift vectors specified directly as a function of position. In ADINA, this technique is termed the SVS (station virtual shift) algorithm, and is described in Section 10.15 for 3-D cracks.

A fundamental advantage of specifying the virtual shift vectors directly as a function of position is that unstructured (free-form) meshes can be used at the crack front.

10.6 Calculation of stress intensity factors using the virtual crack extension method

10.6.1 Overview of theory

The theory in this section is based on material given in the references by Shih, Moran and Nakamura, and Walters, Paulino and Dodds.

In order to calculate stress intensity factors, it is useful to view G as an "interaction integral".

We concentrate on the special case of linear isotropic elastic analysis without thermal or dynamic effects. For this special case,

$W = \frac{1}{2} \tau_{ij} e_{ij}$, and $\tau_{ij} = C_{ijkl} e_{kl}$, where C_{ijkl} is the constitutive tensor.

We now consider an integral closely related to (10.5-1):

$$\tilde{G}(\mathbf{u}^A, \mathbf{u}^B) = \frac{1}{\Delta A_c} \int_V \left(\tau_{ij}^A \frac{\partial u_j^B}{\partial x_k} - \tilde{W}^{A,B} \delta_{ik} \right) \frac{\partial \Delta X_k}{\partial x_i} dV \quad (10.6-1)$$

in which $\tilde{W}^{A,B} = \frac{1}{2} \tau_{ij}^A e_{ij}^B$. This integral measures the interaction of

the stress field corresponding to displacement field \mathbf{u}^A and the displacement gradient field corresponding to displacement field \mathbf{u}^B . The superscript on the displacement field denotes a field label.

Anticipating the developments below, we will use only the symmetric form of (10.6-1), namely

$$\begin{aligned} G(\mathbf{u}^A, \mathbf{u}^B) &= \frac{1}{2} \left(\tilde{G}(\mathbf{u}^A, \mathbf{u}^B) + \tilde{G}(\mathbf{u}^B, \mathbf{u}^A) \right) \\ &= \frac{1}{\Delta A_c} \int_V \left(\frac{1}{2} \left(\tau_{ij}^A \frac{\partial u_j^B}{\partial x_k} + \tau_{ij}^B \frac{\partial u_j^A}{\partial x_k} \right) - W^{A,B} \delta_{ik} \right) \frac{\partial \Delta X_k}{\partial x_i} dV \end{aligned} \quad (10.6-2)$$

in which $W^{A,B} = \frac{1}{4} \left(\tau_{ij}^A e_{ij}^B + \tau_{ij}^B e_{ij}^A \right)$. Because $\tau_{ij} = C_{ijkl} e_{kl}$ in

which $C_{ijkl} = C_{klij}$, $\tau_{ij}^A e_{ij}^B = \tau_{ij}^B e_{ij}^A$ and therefore

$$W^{A,B} = \frac{1}{2} \tau_{ij}^A e_{ij}^B = \frac{1}{2} \tau_{ij}^B e_{ij}^A.$$

Equation (10.6-2) (along with some corrections discussed below) is the basis of the interaction integral method as implemented in ADINA.

In (10.6-2), G is seen to be a function of two displacement fields $\mathbf{u}^A, \mathbf{u}^B$. Clearly, if $\mathbf{u}^A = \mathbf{u}^B = \mathbf{u}$, (10.5-1) is recovered, and in this case, we write

$$G(\mathbf{u}) = G(\mathbf{u}, \mathbf{u}) \quad (10.6-3)$$

Also,

$$G(\mathbf{u}^B, \mathbf{u}^A) = G(\mathbf{u}^A, \mathbf{u}^B) \quad (10.6-4)$$

In addition, $G(\mathbf{u}^A, \mathbf{u}^B)$ is linear in each field $\mathbf{u}^A, \mathbf{u}^B$, so that, for example,

$$G(\alpha \mathbf{u}^A + \beta \mathbf{u}^B, \mathbf{u}^C) = \alpha G(\mathbf{u}^A, \mathbf{u}^C) + \beta G(\mathbf{u}^B, \mathbf{u}^C) \quad (10.6-5)$$

in which $\mathbf{u}^A, \mathbf{u}^B, \mathbf{u}^C$ are three displacement fields and α, β are arbitrary scalars.

The advantage of writing G in the form (10.6-2) is that the fundamental displacement fields used in the definitions of the stress intensity factors (given in Section 10.3) are orthogonal to each other, when introduced into G . In fact, analytically, for a straight crack,

$$G(\mathbf{u}^I) = \frac{1}{E'} \quad (10.6-6a)$$

$$G(\mathbf{u}^{II}) = \frac{1}{E'} \quad (10.6-6b)$$

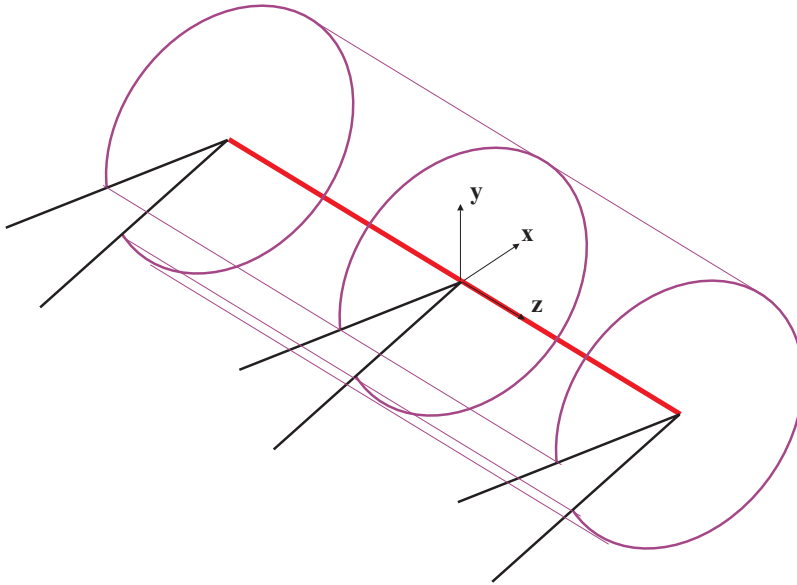
$$G(\mathbf{u}^{III}) = \frac{1}{2\mu} \quad (10.6-6c)$$

$$G(\mathbf{u}^I, \mathbf{u}^{II}) = 0 \quad (10.6-6d)$$

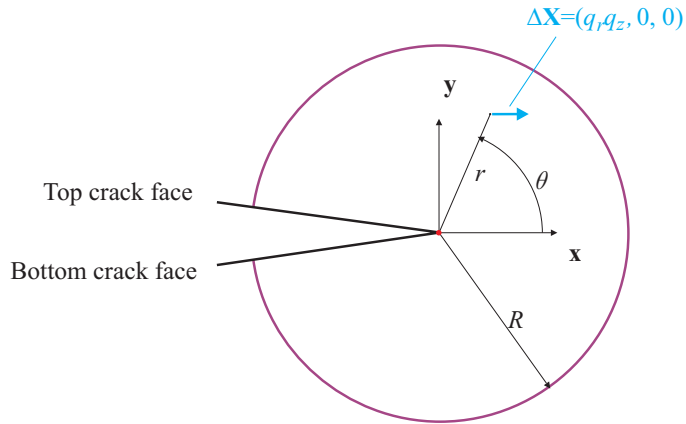
$$G(\mathbf{u}^I, \mathbf{u}^{III}) = 0 \quad (10.6-6e)$$

$$G(\mathbf{u}^{II}, \mathbf{u}^{III}) = 0 \quad (10.6-6f)$$

These results can be obtained by direct computation. In this computation, we assume a straight enclosed crack under plane strain conditions, aligned with the global coordinate system, as shown in Fig. 10.6-1.



a) Virtual shift domain for enclosed crack



b) Side view of virtual shift domain

Figure 10.6-1: Virtual shift domain used to demonstrate orthogonality of fundamental fields in an enclosed crack

We also assume that the virtual shift vector is given by

$$\Delta \mathbf{X} = (q_r q_z, 0, 0) \quad (10.6-7)$$

in which q_r is a function of radius r , with $q_r = 1$ at $r = 0$ and with $q_r = 0$ at $r = R$, and in which q_z is a function of axial coordinate z . Clearly

$$\frac{\partial \Delta \mathbf{X}}{\partial x} = \left(\frac{\partial q_r}{\partial r} q_z \cos \theta, 0, 0 \right) \quad (10.6-8a)$$

$$\frac{\partial \Delta \mathbf{X}}{\partial y} = \left(\frac{\partial q_r}{\partial r} q_z \sin \theta, 0, 0 \right) \quad (10.6-8b)$$

$$\frac{\partial \Delta \mathbf{X}}{\partial z} = \left(q_r \frac{\partial q_z}{\partial z}, 0, 0 \right) \quad (10.6-8c)$$

The corresponding virtual shift area is

$$\Delta A_c = \int q_z dz \quad (10.6-9)$$

By directly substituting the fundamental fields from Section 10.3 into equation (10.6-2) and performing the integrations analytically, the following results are obtained:

$$\begin{bmatrix} G(\mathbf{u}^I) & G(\mathbf{u}^I, \mathbf{u}^{II}) & G(\mathbf{u}^I, \mathbf{u}^{III}) \\ & G(\mathbf{u}^{II}) & G(\mathbf{u}^{II}, \mathbf{u}^{III}) \\ sym & & G(\mathbf{u}^{III}) \end{bmatrix} = \begin{bmatrix} 1/E' & 0 & 0 \\ & 1/E' & 0 \\ sym & & 1/(2\mu) \end{bmatrix} + \frac{\int q_r dr \int \frac{\partial q_z}{\partial z} dz}{\int q_z dz} \begin{bmatrix} 0 & 0 & 0 \\ & 0 & (1+8\nu)/16\mu \\ sym & & 0 \end{bmatrix} \quad (10.6-10)$$

The second term on the right-hand-side disappears when

$\int \frac{\partial q_z}{\partial z} dz = 0$, in other words when the net change of q_z along the

z direction is zero. We will therefore choose q_z so that this condition is satisfied whenever possible. Equations (10.6-6) directly follow from (10.6-10).

For a boundary crack, the same procedure can be used. There are two cases: boundary crack with binormal direction pointing into the material and boundary crack with binormal direction pointing out of the material. These cases are shown in Figure 10.6-2.

And, under the condition $\int \frac{\partial q_z}{\partial z} dz = 0$, the results

$$\begin{bmatrix} G(\mathbf{u}^I) & G(\mathbf{u}^I, \mathbf{u}^{II}) & G(\mathbf{u}^I, \mathbf{u}^{III}) \\ & G(\mathbf{u}^{II}) & G(\mathbf{u}^{II}, \mathbf{u}^{III}) \\ sym & & G(\mathbf{u}^{III}) \end{bmatrix} = \begin{bmatrix} 1/(2E') & 0 & 0 \\ & 1/(2E') & 0 \\ sym & & 1/(4\mu) \end{bmatrix} \quad (10.6-11)$$

are obtained, regardless of whether the binormal points into or out of the material.

Considering the hoop stress, thermal, pressure and dynamic corrections to G , the thermal and dynamic corrections do not apply because the fundamental fields only apply to linear static analysis without thermal effects. The hoop stress correction to (10.6-2) can be written

$$G_{hoop}(\mathbf{u}^A, \mathbf{u}^B) = \frac{1}{\Delta A_c} \int_V \left(\frac{1}{2} \left(\tau_{xx}^A \frac{\partial u_x^B}{\partial x} + \tau_{xx}^B \frac{\partial u_x^A}{\partial x} \right) - W^{A,B} \right) \Delta X_2 dV \quad (10.6-12)$$

The pressure correction can be written

$$G_{pr}(\mathbf{u}^A, \mathbf{u}^B) = -\frac{1}{\Delta A_c} \int_S \frac{1}{2} \left(t_i^A \frac{\partial u_i^B}{\partial x_j} + t_i^B \frac{\partial u_i^A}{\partial x_j} \right) \Delta X_j dS \quad (10.6-13)$$

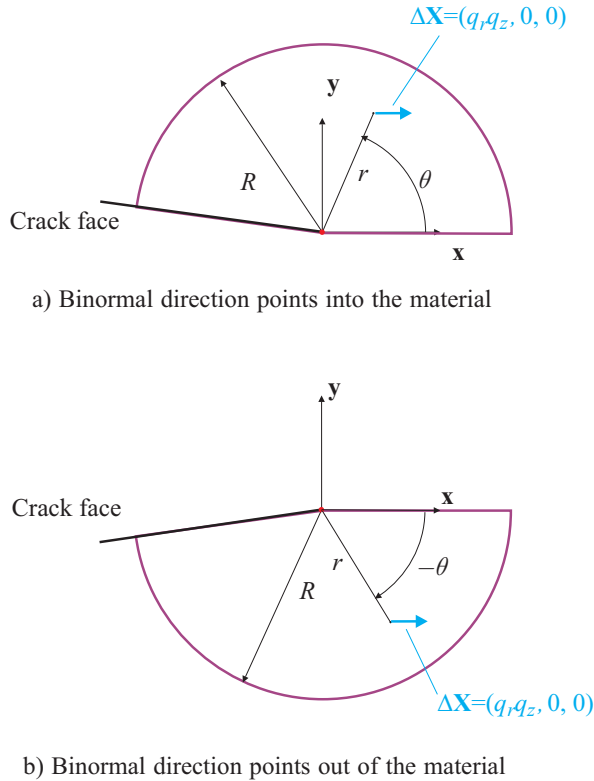


Figure 10.6-2: Virtual shift domains used to demonstrate orthogonality of fundamental fields in boundary cracks

There is an issue when calculating (10.6-13) for the fundamental fields \mathbf{u}^I , \mathbf{u}^{II} , \mathbf{u}^{III} because the tractions for the fundamental fields are unknown. Following the procedure in the reference by Walters, Paulino and Dodds, we assume that:

- The fundamental fields \mathbf{u}^I , \mathbf{u}^{II} , \mathbf{u}^{III} apply without modification when there is a pressure load applied to the crack
- The traction for the fundamental fields is zero.

10.6.2 Calculation of stress intensity factors

We use G in the interaction integral form (equation (10.6-2) and also the corrections from equations (10.6-12) to (10.6-13)) to calculate the stress intensity factors, as follows.

Let \mathbf{u} be the displacement field from the finite element solution. Then (10.3-7) can be rewritten as

$$\mathbf{u}^R = \mathbf{u} - K_I \mathbf{u}^I - K_{II} \mathbf{u}^{II} - K_{III} \mathbf{u}^{III} \quad (10.6-14)$$

Hence

$$\begin{aligned} G(\mathbf{u}^R) &= G(\mathbf{u}) - K_I G(\mathbf{u}, \mathbf{u}^I) - K_{II} G(\mathbf{u}, \mathbf{u}^{II}) - K_{III} G(\mathbf{u}, \mathbf{u}^{III}) \\ &\quad + K_I^2 G(\mathbf{u}^I) + 2K_I K_{II} G(\mathbf{u}^I, \mathbf{u}^{II}) + 2K_I K_{III} G(\mathbf{u}^I, \mathbf{u}^{III}) \\ &\quad + K_{II}^2 G(\mathbf{u}^{II}) + 2K_{II} K_{III} G(\mathbf{u}^{II}, \mathbf{u}^{III}) \\ &\quad + K_{III}^2 G(\mathbf{u}^{III}) \end{aligned} \quad (10.6-15)$$

Now we choose K_I, K_{II}, K_{III} to minimize $G(\mathbf{u}^R)$. The result is

$$\begin{bmatrix} G(\mathbf{u}^I) & G(\mathbf{u}^I, \mathbf{u}^{II}) & G(\mathbf{u}^I, \mathbf{u}^{III}) \\ & G(\mathbf{u}^{II}) & G(\mathbf{u}^{II}, \mathbf{u}^{III}) \\ sym & & G(\mathbf{u}^{III}) \end{bmatrix} \begin{bmatrix} K_I \\ K_{II} \\ K_{III} \end{bmatrix} = \begin{bmatrix} G(\mathbf{u}, \mathbf{u}^I) \\ G(\mathbf{u}, \mathbf{u}^{II}) \\ G(\mathbf{u}, \mathbf{u}^{III}) \end{bmatrix} \quad (10.6-16)$$

The same result is obtained using the unsymmetrized \tilde{G} , because the minimization of $\tilde{G}(\mathbf{u}^R)$ gives the matrix equation

$$\begin{aligned}
 & \begin{bmatrix} 2\tilde{G}(\mathbf{u}^I) & \tilde{G}(\mathbf{u}^I, \mathbf{u}^{II}) + \tilde{G}(\mathbf{u}^{II}, \mathbf{u}^I) & \tilde{G}(\mathbf{u}^I, \mathbf{u}^{III}) + \tilde{G}(\mathbf{u}^{III}, \mathbf{u}^I) \\ & 2\tilde{G}(\mathbf{u}^{II}) & \tilde{G}(\mathbf{u}^{II}, \mathbf{u}^{III}) + \tilde{G}(\mathbf{u}^{III}, \mathbf{u}^{II}) \\ \text{sym} & & 2\tilde{G}(\mathbf{u}^{III}) \end{bmatrix} \begin{bmatrix} K_I \\ K_{II} \\ K_{III} \end{bmatrix} \\
 &= \begin{bmatrix} \tilde{G}(\mathbf{u}, \mathbf{u}^I) + \tilde{G}(\mathbf{u}^I, \mathbf{u}) \\ \tilde{G}(\mathbf{u}, \mathbf{u}^{II}) + \tilde{G}(\mathbf{u}^{II}, \mathbf{u}) \\ \tilde{G}(\mathbf{u}, \mathbf{u}^{III}) + \tilde{G}(\mathbf{u}^{III}, \mathbf{u}) \end{bmatrix} \\
 & \hspace{15em} (10.6-17)
 \end{aligned}$$

Only the symmetric part of \tilde{G} appears in (10.6-17), hence (10.6-16) and (10.6-17) are the same equation. Therefore it is easier to simply define G as the symmetric part of \tilde{G} , then employ G throughout.

10.6.3 Discussion

- This procedure is very closely related to the interaction integral method described in, for example, the reference by Shih and Asaro and the reference by Walters, Paulino and Dodds. In these references, the fields \mathbf{u}^I , \mathbf{u}^{II} , \mathbf{u}^{III} are referred to as auxiliary fields.
- Since the interaction integrals involving only the fundamental fields (e.g. $G(\mathbf{u}^I)$, $G(\mathbf{u}^I, \mathbf{u}^{II})$, etc.) are analytically known, either the analytical values of these integrals can be used in (10.6-16), or the numerical approximations of these integrals can be used.
For example, in the references by Shih, Moran and Nakamura and Walters, Paulino and Dodds, the analytical values are used. However, in ADINA, the numerical approximations are used. In this way, the same numerical integration procedure is used on both sides of (10.6-16).
- The calculation of stress intensity factors using the virtual crack extension method is implemented in ADINA using the SVS algorithm, as discussed in Section 10.15.

10.7 Fracture control parameters

- Here and below, we describe the input for fracture mechanics in terms of the command-line commands of the AUI. This input can also be entered using the AUI user interface. See the Command Reference Manual for ADINA for detailed information about the commands.

MASTER command: The FRACTURE parameter of the MASTER command must be set to YES:

```
MASTER . . . FRACTURE=YES
```

FRACTURE command: The FRACTURE command specifies the type of fracture analysis:

```
FRACTURE  TECHNIQUE  METHOD  DIMENSION  TYPE,  
          PRESSURE  TEMPERATURE  DYNAMIC
```

TECHNIQUE: TECHNIQUE is always set to STANDARD.

METHOD: METHOD can be set to the following values:

LINE-CONTOUR: The J-integral is calculated using line contours. This feature is available only for 2-D cracks.

VIRTUAL-CRACK-EXTENSION: The J-integral is calculated using the NVS method for virtual crack extension. The virtual shifts are expressed in terms of nodal point virtual shifts.

BOTH: Same as using LINE-CONTOUR and also VIRTUAL-CRACK-EXTENSION.

SVS: The J-integral and the stress intensity factors K_I , K_{II} , K_{III} are calculated using the SVS method for virtual crack extension. The virtual shifts are evaluated at crack advance stations. This feature is available only for 3-D cracks.

The default is VIRTUAL-CRACK-EXTENSION.

DIMENSION: DIMENSION=2 for 2-D models and =3 for 3-D

models. The default is 2.

TYPE: TYPE specifies whether the crack is stationary or propagating (default is stationary).

PRESSURE, TEMPERATURE, DYNAMIC: PRESSURE, TEMPERATURE, DYNAMIC specify whether or not to compute the pressure, temperature and dynamic corrections when calculating virtual shifts. The default is to calculate the corrections.

Other commands

Other commands are discussed in the sections listed below:

2-D cracks

<i>Geometry and meshing:</i>	Section 10.8
<i>Method:</i>	
Line contour:	Section 10.9
NVS virtual shift:	Section 10.10
<i>Type:</i>	
Propagating:	Section 10.11

3-D cracks

<i>Geometry and meshing, general comments:</i>	Section 10.12
<i>Meshing using CRACK-M features:</i>	Section 10.13
<i>Method:</i>	
NVS virtual shift:	Section 10.14
SVS virtual shift:	Section 10.15

10.8 2-D crack geometry and meshing

10.8.1 Overview

- The crack line or surface can be located on the boundary of the finite element model (symmetric specimen and loading) or inside the finite element model (asymmetric specimen and/or loading). However, for crack propagation analysis, the specimen and loading must be symmetric.
- Fig. 10.8-1 shows geometries of 2-D models. The crack can be

located inside the finite element model (Fig. 10.8-1(a)) or on the boundary of the model (Fig. 10.8-1(b)). When the crack is located on the boundary of the model, it is recommended that the crack line be parallel to the Y axis, since some of the solution procedures described below assume that the crack line is parallel to the Y axis.

Fig. 10.8-1 also shows the “line of self-similar crack advance”. As seen below, it can be advantageous to use a mesh in which nodes are placed along this line. Of course, when the crack is located on the boundary of the model, then this line also is on the boundary of the model.

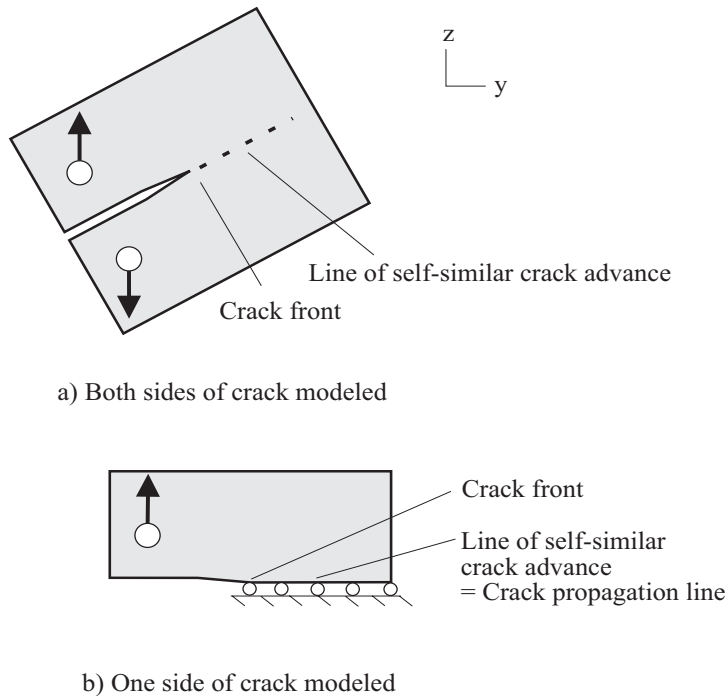
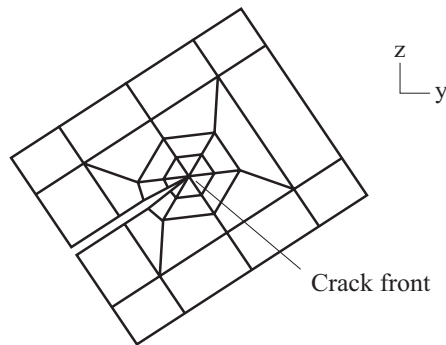


Figure 10.8-1: 2-D fracture geometries

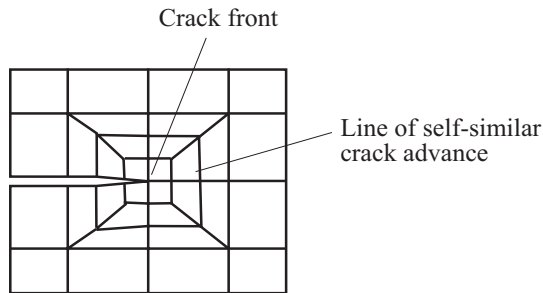
- The terminology “line of self-similar crack advance” is used instead of “crack propagation line” to emphasize that the crack might not actually propagate along this line under mixed mode conditions. However, for symmetric geometries and loadings, the

line of self-similar crack advance is the same as the crack propagation line.

- Illustrative 2-D meshes are shown in Fig 10.8-2. Notice that, when both sides of the crack are modeled, there might or might not be nodes in the line of self-similar crack advance. The input corresponding to these possibilities is described below.

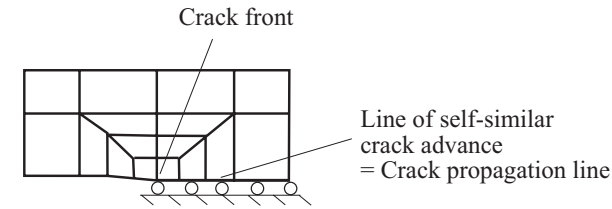


- a) Both sides of crack modeled,
no nodes on line of self-similar
crack advance



- b) Both sides of crack modeled,
nodes on line of self-similar
crack advance

Figure 10.8-2: Meshes for 2-D fracture geometries



c) One side of crack modeled

Figure 10.8-2: Continued

10.8.2 Meshing recommendations

- In linear elastic analysis, the variation of the stress and strain fields in the proximity of the crack tip is characterized by a $\frac{1}{\sqrt{r}}$ singularity where r is the distance to the crack tip location (see Section 10.3). Such a variation can be modeled using collapsed quadrilateral elements with collapsed nodes at the crack tip location, and midside nodes at the 1/4 points. True triangular elements with midside nodes at the 1/4 points can also be used. Fig 10.8-3 shows some examples.

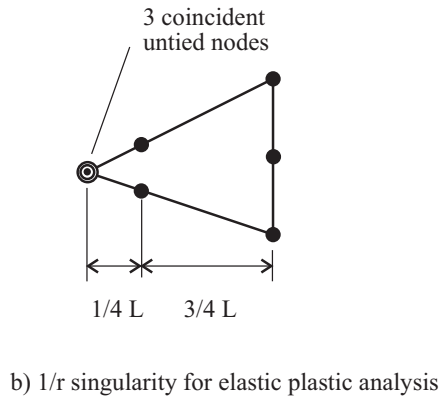
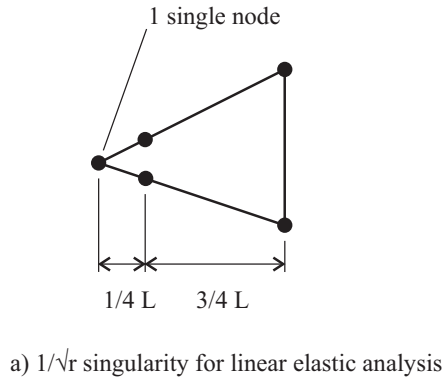


Figure 10.8-3: Element configurations for the modeling of stress field singularities

The SINGULAR command can be used to shift the element nodes near the crack tip to the quarter-points. This command has the following forms:

```
SINGULAR POINT Q-POINT
pointi
```

The nodes at points `pointi` are considered to be at the crack tip. The adjacent mid-side nodes are shifted to the quarter-points.

Note that the SINGULAR commands shift the nodes either to or from the quarter-points, depending upon parameter Q-POINT. The

default is to shift the nodes to the quarter-points.

- For the analysis of elastic-plastic structures, collapsed quadrilateral elements with 3 coincident but untied nodes at the crack tip location are recommended, in order to reproduce a $1/r$ variation of the stress/strain field.
- Typical meshes for the evaluation of crack tip blunting, or for the calculation of local crack opening parameters such as the CTOD, are shown in Fig. 10.8-4. These meshes can be generated using the specific meshing features of the AUI. In particular, the GSURFACE command has several parameters used to generate this type of mesh.

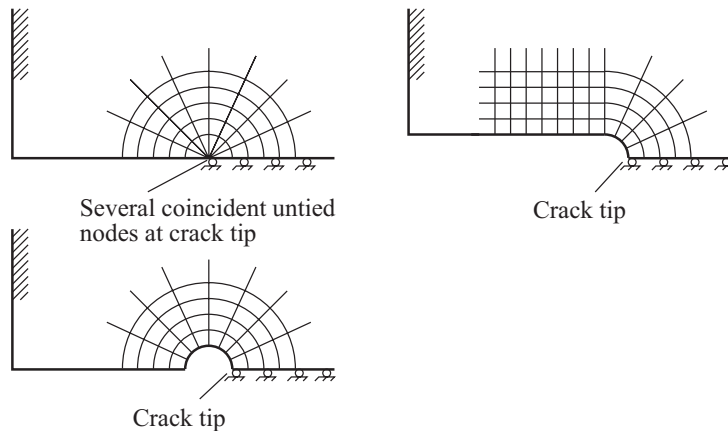


Figure 10.8-4: Meshes for the analysis of blunting and local crack opening criteria

- A typical mesh for 2-D crack propagation analysis is shown in Fig. 10.8-5. Quadrilateral elements with any number of nodes can be used. To ensure good results, the mesh in the area where the crack propagates should be made of a regular density of elements with regular shapes.

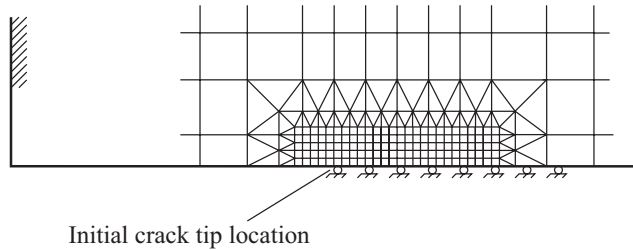


Figure 10.8-5: Mesh used for crack propagation

10.8.3 Definition of the crack propagation line, no nodes on the line of self-similar crack advance

- When there are no nodes on the line of self-similar crack advance, then only the crack front node needs to be specified. This specification is done with the AUI commands

```
CRACK-PROPAGATION NODES  NCRACK=1
node-1i
```

or

```
CRACK-PROPAGATION POINT
pointi
```

- In the above cases, only one data input line of the CRACK-PROPAGATION command is used.

10.8.4 Definition of the crack propagation line, nodes on the line of self-similar crack advance

- When there are nodes on the line of self-similar crack advance, then the crack front node and also the line of self-similar crack advance can be specified. This specification is done with the AUI commands

```
CRACK-PROPAGATION NODE  NCRACK>1
node-1i node-2i ... node-NCRACKi
```

where node-1_i is the crack front node, node-2_i is the next corner node on the line of self-similar crack advance,

or

```
CRACK-PROPAGATION LINE  
linei front-pointi
```

where `linei` is the geometry line of self-similar crack advance and `front-pointi` is the geometry point at the crack front.

- In the above cases, only one data input line of the CRACK-PROPAGATION command is used.
- When only one side of the crack is modeled, ADINA automatically generates a zero prescribed displacement in the z direction for the crack front node and for each node on the line of self-similar crack advance. (If skew systems are used, the zero prescribed displacement is generated in the skew c direction.)
- It will be seen that there are more fracture options available when there are nodes on the line of self-similar crack advance. Thus the model should contain nodes on the line of self-similar crack advance when possible.

10.9 Definition and use of line contours for 2-D cracks

10.9.1 Definition of line contours

- In ADINA, the line contour Γ of Eq. (10.4-1) is defined by a series of adjoining segments passing through the elements located on the contour (see Fig. 10.9-1). The integration of the J-integral along each segment is performed numerically using the value of the variables at the integration points which define the segment.

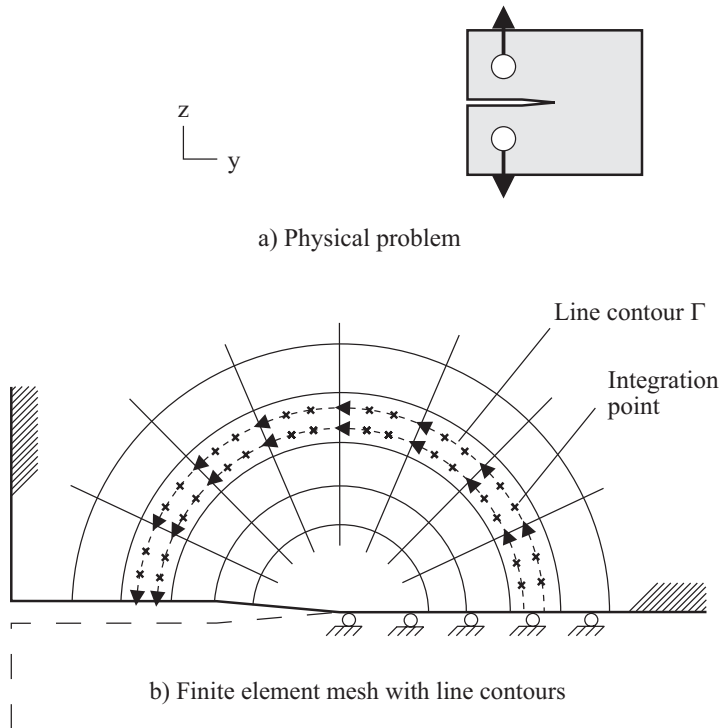


Figure 10.9-1 Definition of line contours in a two-dimensional finite element model.

- As many contours as desired can be requested in ADINA. Usually, several contours are used simultaneously in order to assess the J-integral path-independence.
- 2-D elements with 2×2 or 3×3 Gauss integration can be used. True triangular elements (with triangular integration) cannot be used. All the elements on one contour must have the same integration order. If 2×2 Gauss integration is used, the J-integral is evaluated simultaneously along two line contours (see Fig. 10.9-2). If 3×3 Gauss integration is used, only one J-integral is evaluated, using a middle path through each element (see Fig. 10.9-2).

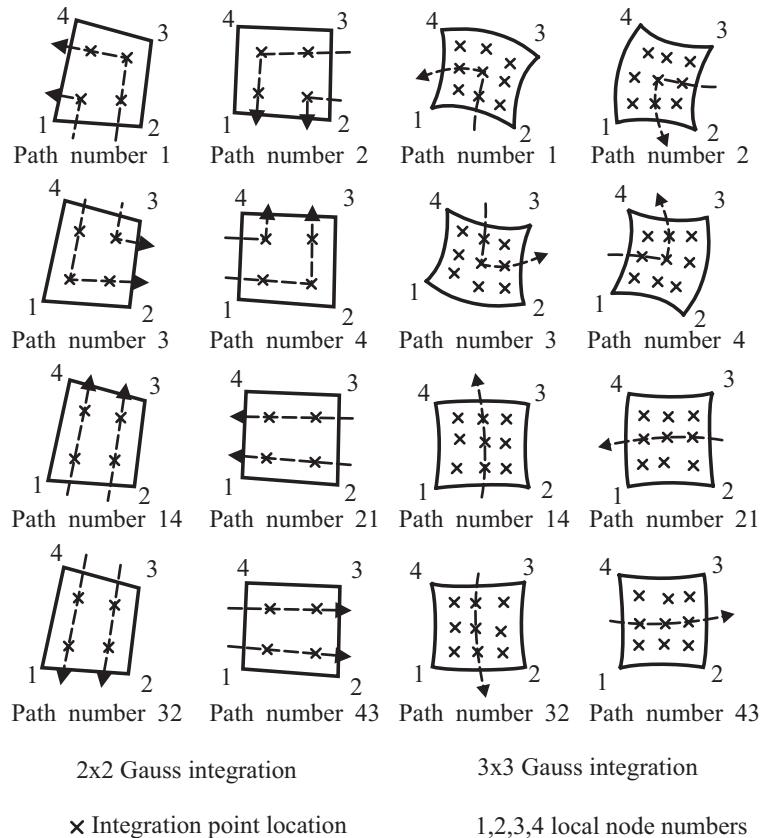


Figure 10.9-2: Convention for the description of element paths

- In order to describe a contour directly in terms of the element numbers, the element numbers (and the group numbers if necessary) must be input in a sequence following the line contour from one end to the other. For the first and last elements on the contour, at least one face must coincide with the physical boundary of the model. The AUI command is

J-LINE ELEMENT
 $\text{element}_i \quad \text{group}_i$

- The AUI provides commands for the automatic generation of line contours.

By geometry point and radius:

J-LINE POINT ... POINT RADIUS

The line contour is defined by a series of elements intersected by the circle with origin at geometry point POINT and radius RADIUS.

By geometry point and ring:

J-LINE RING ... NRING POINT

The line contour is defined by a series of elements which have ring number NRING, starting from geometry point POINT. Elements with ring number 1 are those connected to the crack tip, elements with ring number 2 are those connected to ring number 1, etc.

10.9.2 Output

Line contour variables

Each line contour outputs the following information to the porthole file:

LINE_J-INTEGRAL_AVERAGE,
LINE_J-INTEGRAL_DIFFERENCE.

If 2×2 Gauss integration is used, LINE_J-INTEGRAL_AVERAGE is the average of the two J-integrals evaluated from the equations in Section 10.4 and LINE_J-INTEGRAL_DIFFERENCE is the difference between these values. If 3×3 Gauss integration is used, LINE_J-INTEGRAL_AVERAGE is the value of the J-integral and LINE_J-INTEGRAL_DIFFERENCE is zero.

10.10 Definition and use of virtual shifts for 2-D cracks

- The NVS method of virtual crack extension is always used for 2-D cracks.

- A virtual shift requires two items to be specified:
 - ▶ The domain of the virtual shift
 - ▶ The virtual shift vector

Each of these items is described in more detail below.

- The virtual shift vector field described in Section 10.5 is specified by virtual shift vectors at the nodal points. Thus the virtual shift domain is specified in terms of the node numbers.
- Figure 10.10-1 shows a virtual shift domain. Within zone I, all of the corner nodes are shifted by the same virtual shift vector. Since zone I contains the crack front, the crack grows slightly, and the crack area increases by amount ΔA_c . Within zone III, none of the corner nodes are shifted, thus the zone is unchanged by the virtual shift.

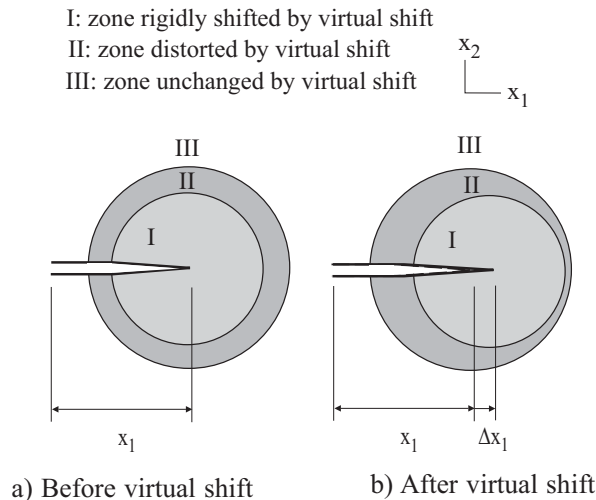


Figure 10.10-1: The virtual crack extension method in two-dimensional analysis

- Corner nodes in the virtual shift are shifted by the full value of the virtual shift vector. Mid-side nodes are shifted by vectors interpolated from the corresponding corner nodes, so that a mid-

side node between two corner nodes in the virtual shift is shifted by the full value of the virtual shift vector, and a mid-side node halfway between a corner node in the virtual shift and an unshifted corner node is shifted by half the value of the virtual shift vector.

- As many virtual shifts as desired can be specified.
- It is recommended that several virtual shifts of increasing domain size be used. Each virtual shift should enclose the next smaller virtual shift. The virtual shift domain size should increase equally in all directions from the plane perpendicular to the crack front.

10.10.1 Domain of the virtual shift

By node numbers: The most basic way to specify a virtual shift is to specify its corner (vertex) node numbers. In Fig. 10.10-2, corner nodes in the virtual shift are shown as solid circles and other nodes are unmarked. All elements with all corner nodes in the virtual shift are in zone I, and are rigidly translated by the virtual shift. All elements with only some corner nodes in the virtual shift are in zone II, and are distorted by the virtual shift. All other elements are in zone III.

The AUI command is

```
J-VIRTUAL-SHIFT  NODE  
nodei
```

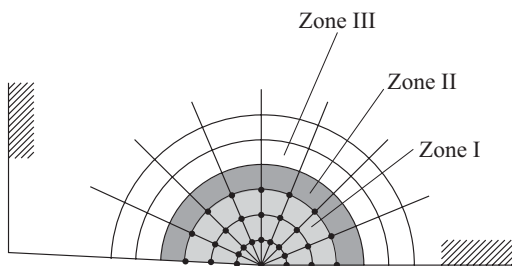


Figure 10.10-2: Description of a virtual shift by nodes

By element numbers: A virtual shift domain can be defined in terms of element numbers. In this case, each element specified is

in zone I and other elements are in zones II and III. The basic idea is shown in Fig. 10.10-3.

The AUI command is

```
J-VIRTUAL-SHIFT ELEMENT  
elementi groupi
```

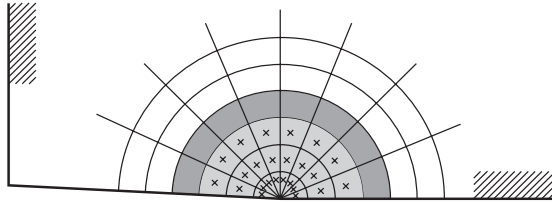


Figure 10.10-3: Description of virtual shifts by element numbers

By line or surface numbers: The nodes in a virtual shift domain can be selected using geometry lines or surfaces.

The AUI commands are

```
J-VIRTUAL-SHIFT LINE  
linei
```

and

```
J-VIRTUAL-SHIFT SURFACE  
surfacei
```

All nodes lying on the given lines or surfaces are in zone I.

By point and radius: A virtual shift domain can be defined using a sphere (point at center and radius).

The AUI command is

```
J-VIRTUAL-SHIFT POINT ... POINT RADIUS
```

For 2-D analysis, since the intersection of a sphere with origin in the y-z plane and the y-z plane is a circle, this command can be

thought of as defining a circle with given center and radius. All nodes within the circle are in zone I. This is shown in Fig. 10.10-4. Note that usually the center is at the crack tip, but this is not a requirement; the center can be at any geometry point, as long as the circle contains the crack tip.

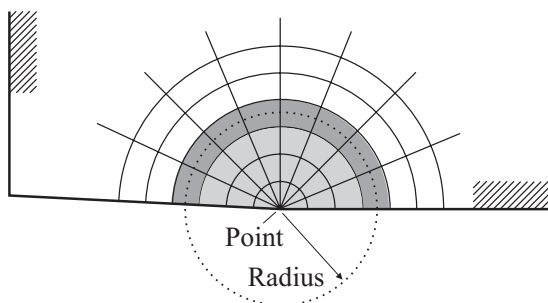


Figure 10.10-4: Description of a virtual shift by point and radius

By rings: A virtual shift domain can be defined in terms of rings. Here the origin of the ring and the number of rings are specified, as shown in Fig. 10.10-5 for 2-D analysis.

The AUI command is

```
J-VIRTUAL-SHIFT RING ... RING-TYPE RING-NUMBER
namei
```

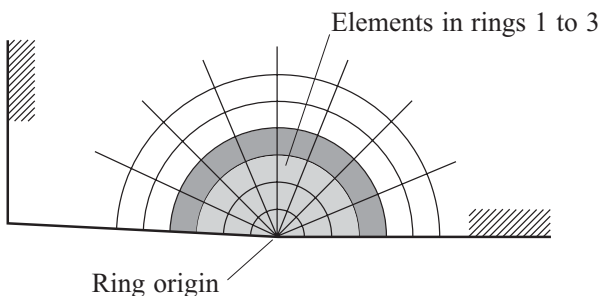


Figure 10.10-5: Description of a virtual shift by ring number

The origin of the ring can either be a single node or a set of nodes. These origin nodes are selected by parameters RING-TYPE and the

data input lines:

RING-TYPE=POINT: The origin nodes are selected by geometry points, so that each “name_i” is a geometry point.

RING-TYPE=LINE: The origin nodes are selected by geometry lines, so that each “name_i” is a geometry line.

RING-TYPE=SURFACE: The origin nodes are selected by geometry surfaces, so that each “name_i” is a geometry surface.

RING-TYPE=NODE: The origin nodes are selected by nodes, so that each “name_i” is a node.

The number of rings is selected by RING-NUMBER. RING-NUMBER=0 just selects the origin nodes, RING-NUMBER=1 selects the origin nodes plus nodes in elements connected to the origin nodes, RING-NUMBER=2 selects nodes in ring 1 plus nodes in elements connected to nodes in ring 1, etc.

10.10.2 Virtual shift vector

Parameters VECTOR, VX, VY, VZ, N3DSH of the above J-VIRTUAL-SHIFT commands are used to define the virtual shift vector, as follows:

VECTOR=INPUT: The shift vector is defined directly using parameters VX, VY, VZ. Note that both the magnitude and direction of this vector are used. This vector should lie in the line or plane of self-similar crack advance.

VECTOR=AUTOMATIC: This option can be used only when there are nodes on the line of self-similar crack advance and when these nodes are specified in the CRACK-PROPAGATION command. The virtual shift vector lies on the line of self-similar crack advance and has magnitude of 1/100 the distance between the crack tip and the next corner node along this line.

10.10.3 Output

Virtual shift variables

Each virtual shift prints the following information, and outputs the following information to the porthole file. This information is accessed in the AUI using the given variable names.

J-PARAMETER_1:
(obsolete)

J-PARAMETER_2:
J-integral without thermal, pressure or dynamic corrections,
equal to $G_I + G_{hoop}$.

J-PARAMETER_3:
J-integral with temperature, pressure and dynamic
corrections, equal to $G_I + G_{hoop} + G_{th} + G_{pr} + G_{dyn}$.

TEMPERATURE_CORRECTION, PRESSURE_CORRECTION,
DYNAMIC_CORRECTION:
The corrections G_{th} , G_{pr} , G_{dyn}

10.11 Crack propagation for 2-D cracks

- The analysis of crack propagation is available in ADINA for 2-D models. The crack must be planar, on the symmetry line of a symmetric structure submitted to a symmetric loading.
- The "node shift/release" technique used in ADINA combines the shifting and the releasing of the successive crack tip nodes in order to model the propagation of the crack tip through the finite element mesh (see Fig. 10.11-1).
- The "node release-only" technique can also be used, whereby the crack tip node is not shifted but only released. This option provides better convergence when large deformations occur and convergence difficulties arise with the node shift/release technique.

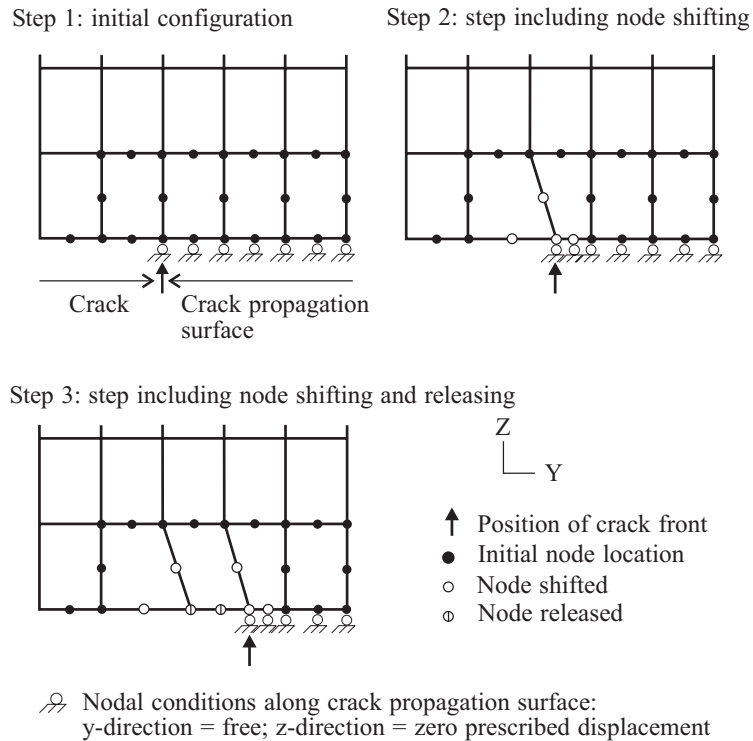


Figure 10.11-1: The node shift / release procedure for crack propagation

Definition of crack advance: The crack advance is the distance that the crack has advanced since the time of solution start (current crack length minus initial crack length).

Definition of the crack propagation surface: The crack propagation surface, which typically corresponds to the ligament in a fracture mechanics test specimen, is defined in ADINA as the set of nodes which may possibly be released when the crack opens. The crack propagation surface must be planar. In two-dimensional models, the crack propagation surface reduces to a line, though still called a surface, and is the same as the line of self-similar crack advance.

The commands in Section 10.10.1 should be used to define the crack propagation surface.

Crack growth control parameter: Different parameters can be used in ADINA for the control of the crack advance.

- Displacement at a node. This displacement is typically chosen to be at the crack tip (to monitor the crack tip opening displacement (CTOD)), at the crack mouth (to monitor the crack mouth opening displacement (CMOD)), or at the point of load application (see Fig. 10.11-2).
- J-integrals with or without thermal, pressure, and dynamic corrections.

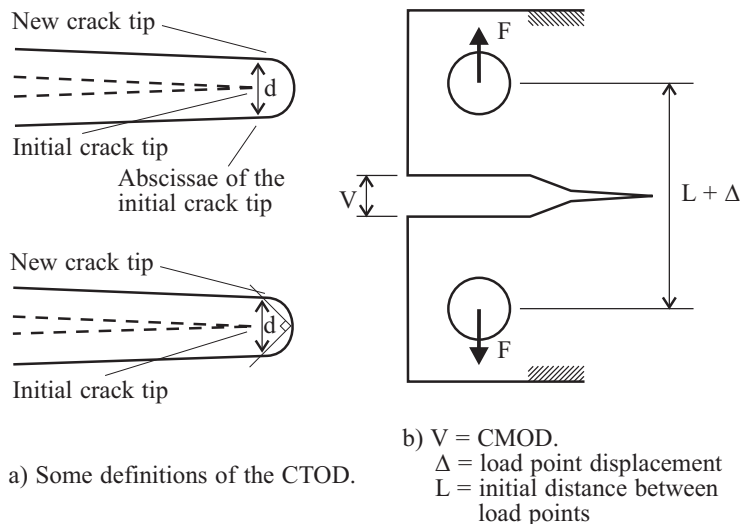


Figure 10.11-2: Definitions of the crack tip opening displacement, crack mouth opening displacement, load point displacement

- If the J-integral is chosen as the crack growth control parameter, the associated virtual shift can be either a spatially fixed shift or a spatially moving shift (see Fig 10.11-3).

A fixed virtual shift can be defined using any of the virtual shift definition commands given in Section 10.10. The virtual shift domain of a fixed virtual shift must be chosen large enough to include the successive crack tip locations at any time/load step of

the analysis.

When a moving virtual shift is requested, ADINA automatically creates a moving virtual shift as the next higher virtual shift number (e.g, if 7 virtual shifts are defined using the commands given above, then the moving virtual shift is shift number 8). The size of the moving virtual shift is defined by the number of rings of elements connected to the crack tip node. ADINA automatically updates the moving virtual shift definition as the crack tip node changes.

Crack resistance curve: In order to evaluate the amount of crack advance from the value of the J-integral or the displacement, a resistance curve must be input. This curve is a material property.

When the crack growth control parameter is a displacement parameter, then this curve relates the value of the displacement parameter to the crack advance. When the crack growth control parameter is the J-integral, then this curve relates the value of the J-integral to the crack advance.

Fig. 10.11-4(a) shows a resistance curve used when the materials are temperature-independent. For each value of the crack growth control parameter, the resistance curve gives the threshold value for crack advance.

For materials with temperature dependent material properties, a set of temperature dependent resistance curves can be input. The temperature at the crack tip node is used with the current value of the crack growth control parameter to evaluate the threshold value for crack advance (see Fig. 10.11-4(b)). Notice that the interpolation is in the horizontal direction.

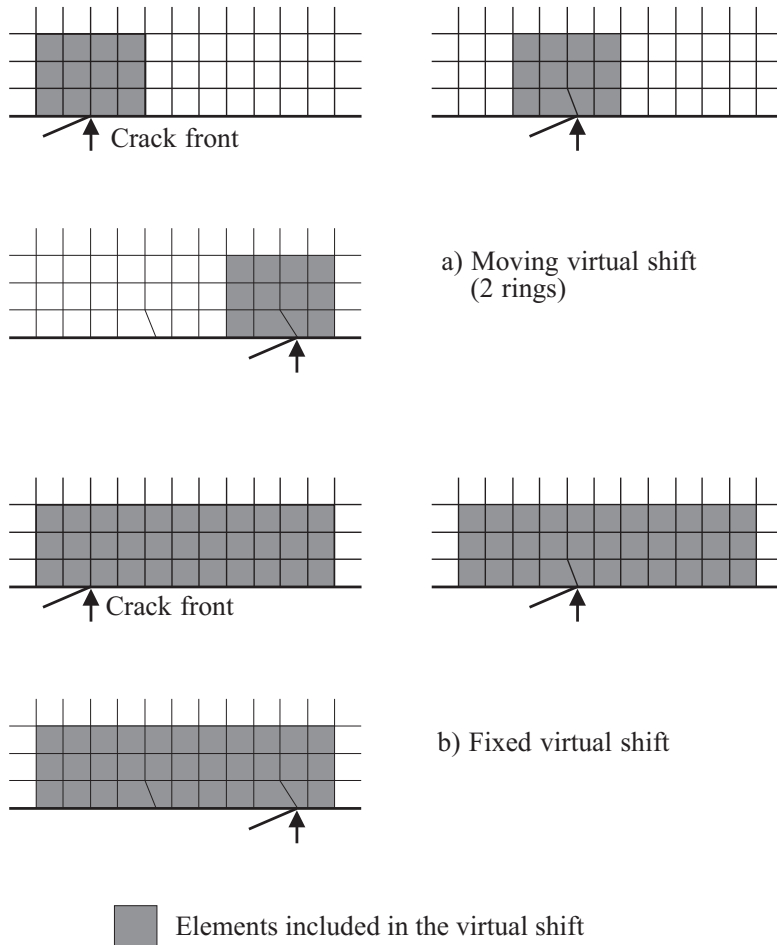
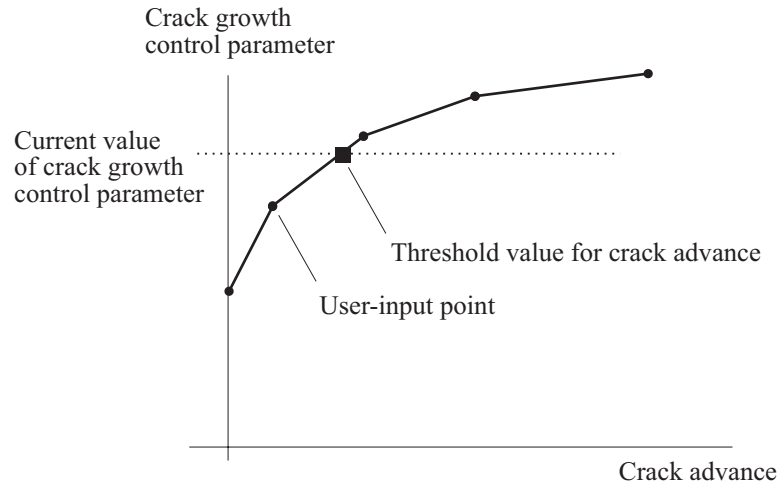
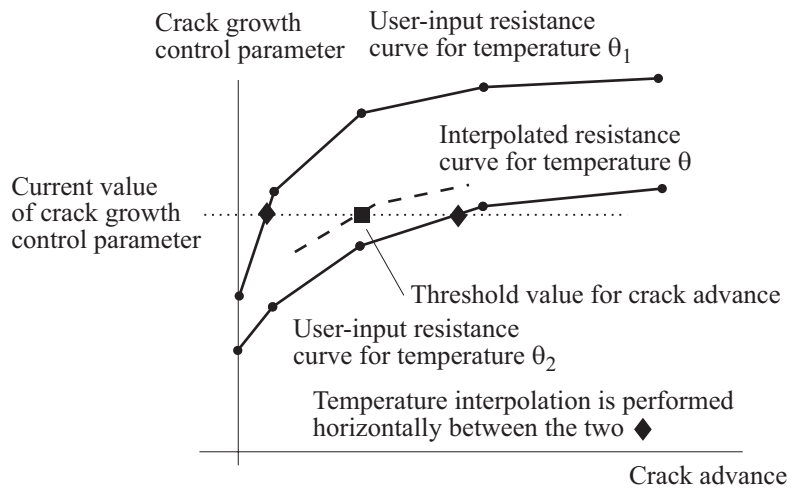


Figure 10-11.3: Moving and fixed virtual shifts in crack propagation analysis



a) Temperature independent



b) Temperature dependent

Figure 10.11-4: Resistance curves

Modeling of the crack advance : At the end of each time step, ADINA computes the crack growth control parameter and computes the threshold crack advance from the resistance curve. If the threshold crack advance is less than the current crack advance,

the crack advance remains unchanged. If the threshold crack advance is greater than the current crack advance, the crack advance is updated using either the node shift/release technique or the node release technique. The idea is shown in Fig. 10.11-5.

The crack advance is considered to take place at the beginning of the next time step. Thus a plot of the crack growth control parameter vs the crack advance will most likely show the crack growth control parameter higher than the critical value for at least a few steps. This mismatch can be reduced by taking smaller time steps.

When the crack growth control parameter is the J-integral, ADINA only computes the crack advance for those time/load steps in which the stresses are printed or saved. Therefore it is recommended that the stresses be printed or saved for every step.

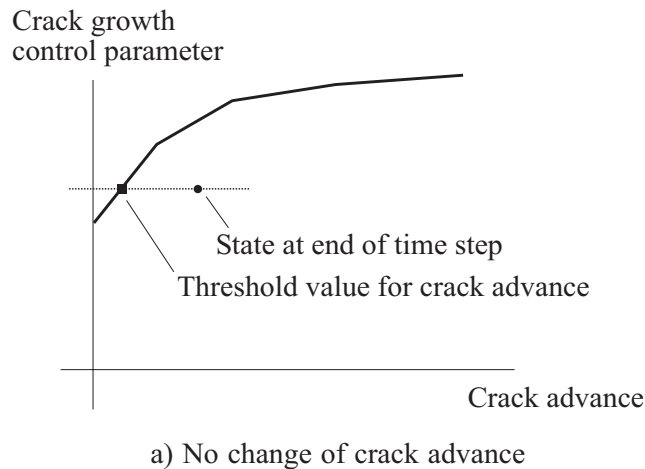


Figure 10.11-5: Calculation of crack advance

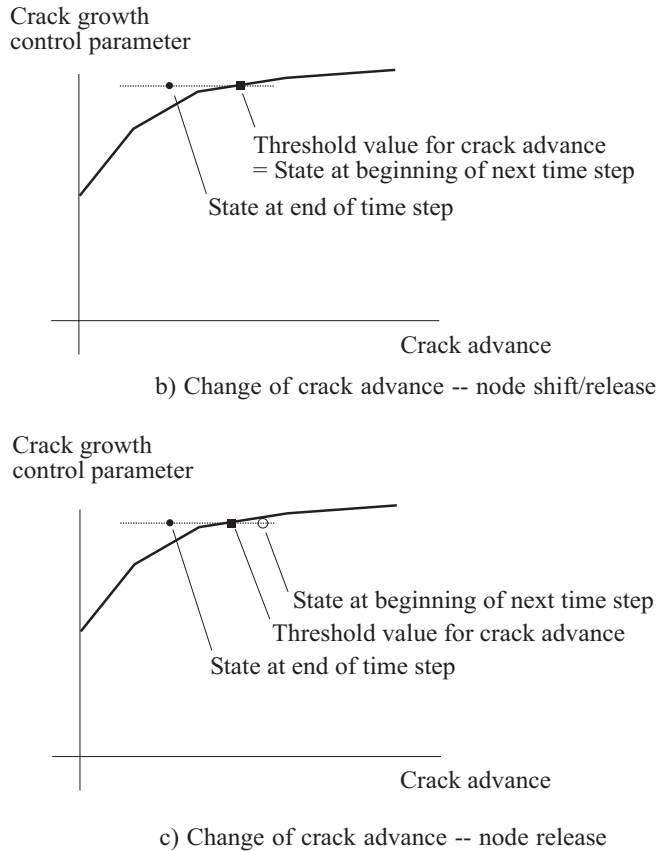


Figure 10.11-5: Continued

Modeling of the crack advance using the node shift/release

technique: When the crack growth control parameter indicates a crack increase, the crack tip node is automatically shifted to a new position (see Fig. 10.11-1 - step 2). When the advance of the crack tip node becomes larger than the width of the element ahead of the crack tip node in the propagation direction (and connected to the crack propagation surface), the crack tip node is released and the next node on the crack propagation surface starts being shifted (Fig. 10.11-1 - step 3). Midside nodes are shifted to remain halfway between the corresponding corner nodes, and are released when the preceding corner node on the crack propagation surface is released.

This procedure does not require any specific element size or

mesh refinement, apart from general considerations on the accuracy of the analysis results. However, when an elastic-plastic material model is used in the shifted elements, the material state is not remapped during shifting. This can cause the solution to be inaccurate.

Quadrilateral elements should be used in the portion of the model including the crack tip and the crack propagation surface, in order to allow the nodes on the crack propagation surface to shift as the crack grows.

Modeling of the crack advance using the node release

technique: When the crack growth control parameter indicates a crack increase, if the crack increase is less than one-half the element length, the crack tip node is not released. If the crack increase is greater than one-half the element length, the crack tip node (and mid-side node, if any) is released. In general, nodes are released so that the difference between the desired crack advance and the actual crack advance (modeled by releasing nodes) is minimized.

Since the element coordinates are never shifted during this procedure, the issue of remapping the material state does not arise.

Pressure loading: In case the faces of the crack are subjected to pressure loads, these loads should be applied to both the initial crack faces and the crack propagation surface. The pressure loads are updated as the nodes are shifted and released.

Input specification:

FRACTURE: The following command specifies a crack propagation analysis:

```
FRACTURE    TYPE=PROPAGATION
```

CRACK-GROWTH: The following command specifies parameters associated with crack propagation:

```
CRACK-GROWTH CONTROL-TYPE  J-VERSION,  
                        R-CURVE  LOCTYPE  DOF  SHIFT-RELEASE,  
                        POINT  NODE
```

If CONTROL-TYPE=FIXED or MOVING, the crack growth control parameter is the J-integral associated with either a fixed or moving virtual shift; J-VERSION specifies whether the J-integral includes corrections. If CONTROL-TYPE=NODAL, the crack growth control parameter is a nodal displacement; parameters LOCTYPE, DOF, POINT, NODE specify the location of the displacement and the displacement direction.

R-CURVE specifies the number of the resistance curve to use. This resistance curve is defined using the R-CURVE command.

SHIFT-RELEASE specifies whether to use the node shift/release method or the node release method.

CRACK-PROPAGATION: The CRACK-PROPAGATION command includes a data input line parameter “nvshft_i”, used as follows:

CONTROL-TYPE=FIXED, nvshft_i is the number of the fixed virtual shift used to control crack growth.

CONTROL-TYPE=MOVING, nvshft_i is the number of rings in the moving virtual shift.

Parameter nvshft_i is specified in the data input lines, as shown:

```
CRACK-PROPAGATION NODES  NCRACK>1
node-1i node-2i ... node-NCRACKi  nvshfti
```

```
CRACK-PROPAGATION LINE
linei  front-pointi  nvshfti
```

R-CURVE: This command defines a set of resistance curves.

```
R-CURVE  NAME  MPOINT
thetai  x1i  y1i  x2i  y2i  xmi  ymi
```

Each data input line specifies the (x,y) points for the resistance curve for a given temperature. MPOINT is the number of points in each resistance curve. In this command, “x” is the crack advance and “y” is the crack growth control parameter. If there are no temperature-dependent materials, then only one curve need be

entered and θ , is not used. When there are temperature-dependent materials, then at least two curves need to be entered. The values of “ x ” must be the same for each resistance curve.

10.12 3-D crack geometry and meshing

10.12.1 Overview

- Fig. 10.12-1 shows geometries of 3-D models. The crack can be located inside the finite element model (enclosed crack) or on the boundary of the model (boundary crack).

Fig. 10.12-1 also shows the “plane of self-similar crack advance”. Of course, when the crack is located on the boundary of the model, then this plane also is on the boundary of the model.

- The mesh layout near the crack front strongly influences the methods that can be used for virtual crack extension. The NVS method for virtual crack extension can only be used provided that the mesh has a specific layout near the crack front. The mesh layout required for the NVS method is discussed in detail in Section 10.14. The SVS method for virtual crack extension can be used for any mesh layout, including free-form mesh layouts, as discussed in detail in Section 10.15.
- 3-D fracture models can be constructed either with the use of the CRACK-M meshing features or without the use of the CRACK-M meshing features. The CRACK-M meshing features are designed to create mapped meshes of prismatic elements near the crack front and free-form meshes away from the crack front. These meshes have the mesh layout required by the NVS method. However, the CRACK-M features must be used in conjunction with the ADINA-M solid modeler. The CRACK-M features are discussed in detail in Section 10.13.

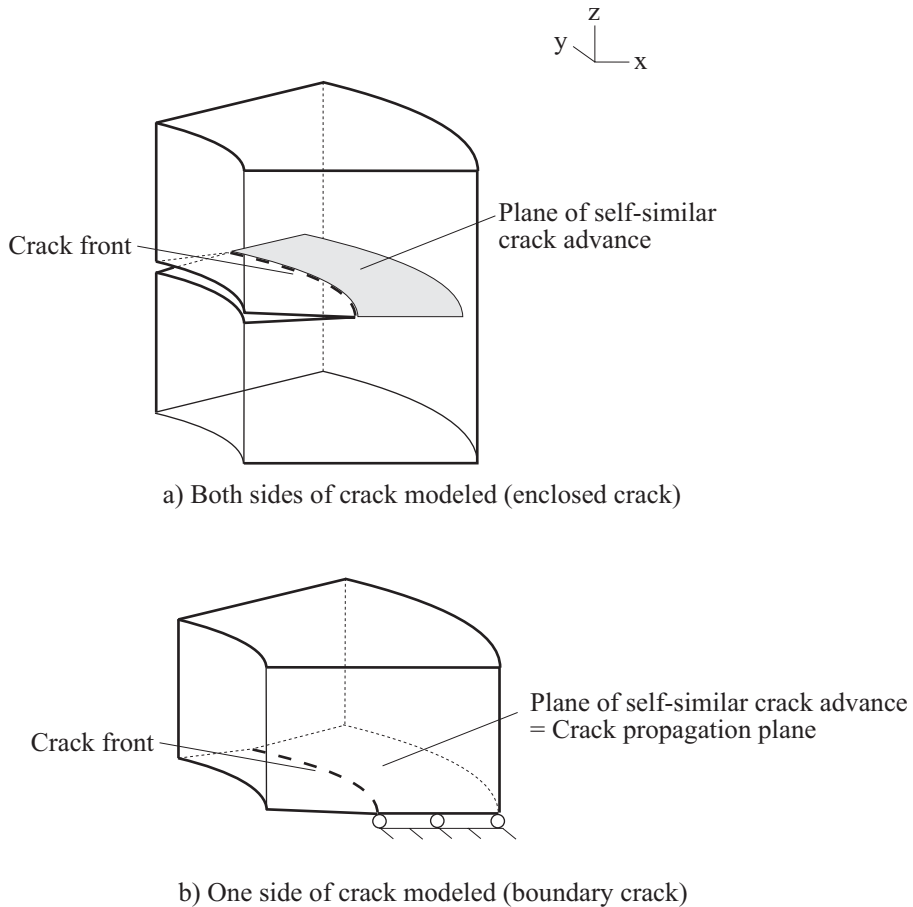


Figure 10.12-1: 3-D fracture geometries

10.12.2 Meshing recommendations

- In linear elastic analysis, the variation of the stress and strain fields in the proximity of the crack tip is characterized by a $\frac{1}{\sqrt{r}}$ singularity where r is the distance to the crack tip location, see Section 10.3. Such a variation can be modeled using collapsed

wedge brick and tetrahedral elements with collapsed nodes at the crack tip location, and midside nodes at the $1/4$ points, as shown in Fig. 10.12-2.

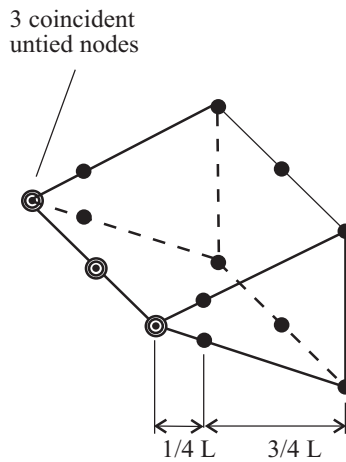
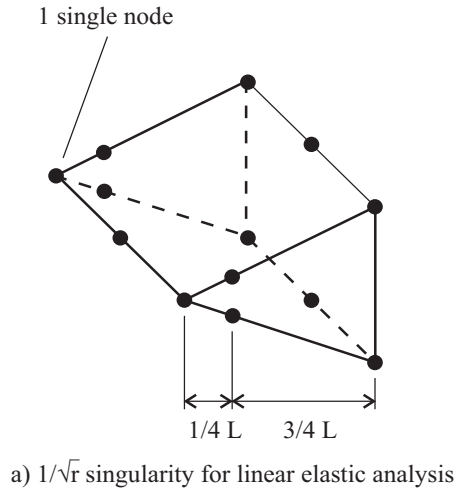
The SINGULAR commands can be used to shift the element nodes near the crack front to the quarter-points.

```
SINGULAR LINE Q-POINT  
linei
```

The nodes on lines line_i are considered to be at the crack front. The adjacent mid-side nodes are shifted to the quarter-points.

Note that the SINGULAR commands shift the nodes either to or from the quarter-points, depending upon parameter Q-POINT. The default is to shift the nodes to the quarter-points.

- The SINGULAR commands should not be used when using the CRACK-M modeling features. Use the CRACK-M Q-POINT command instead. The SINGULAR commands also should not be used when using the SVS method for virtual crack extension, since the CRACK-SVS command includes the option of shifting nodes to the quarter-points.
- For the analysis of elastic-plastic structures, collapsed quadrilateral elements with 3 coincident but untied nodes at the crack tip location are recommended, in order to reproduce a $1/r$ variation of the stress/strain field. In 3-D analysis, wedge elements with coincident but untied nodes on the crack front will reproduce the $1/r$ stress/strain singularity (see Fig. 10.12-2).

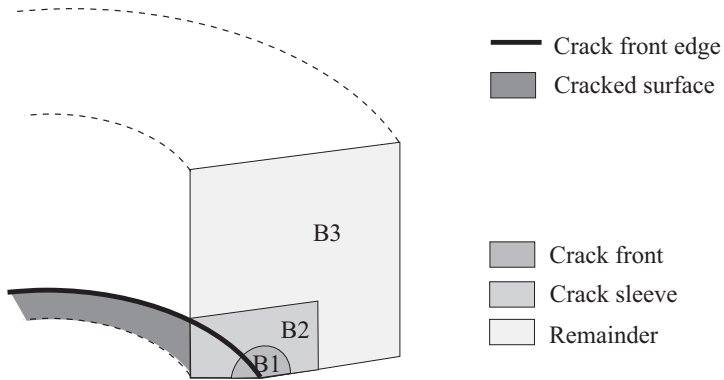


b) $1/r$ singularity for elastic plastic analysis

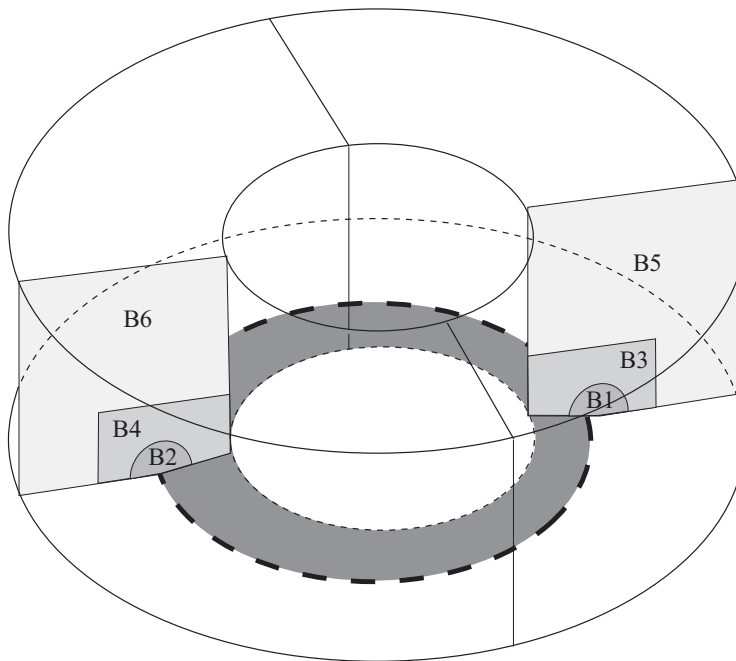
Figure 10.12-2: Element configurations for the modeling of stress field singularities

10.13 3-D crack geometry and meshing using CRACK-M

- Using the CRACK-M features, complex 3-D models with embedded cracks can be easily constructed. The ADINA-M geometric modeler must be used.
- The CRACK-M meshing features are designed to create mapped meshes of prismatic elements near the crack front and free-form meshes away from the crack front.
- Meshes created using the CRACK-M features can be used in conjunction with either the NVS or SVS methods for virtual crack extension.
- The basic ideas are shown in Figs 10.13-1 and 10.13-2. Fig 10.13-1 shows models in which the crack is on the boundary and Fig 10.13-2 shows models in which the crack is in the interior (enclosed crack).

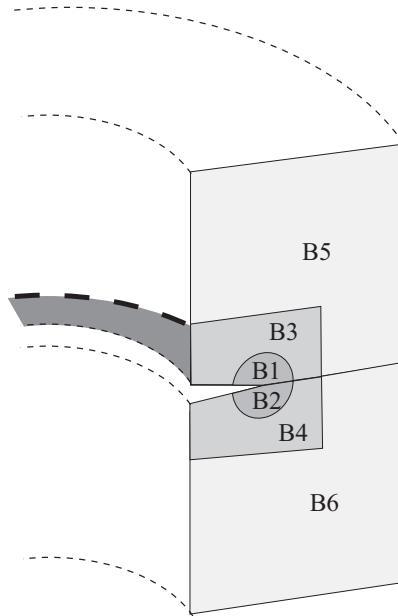


a) Crack front edge is an open loop



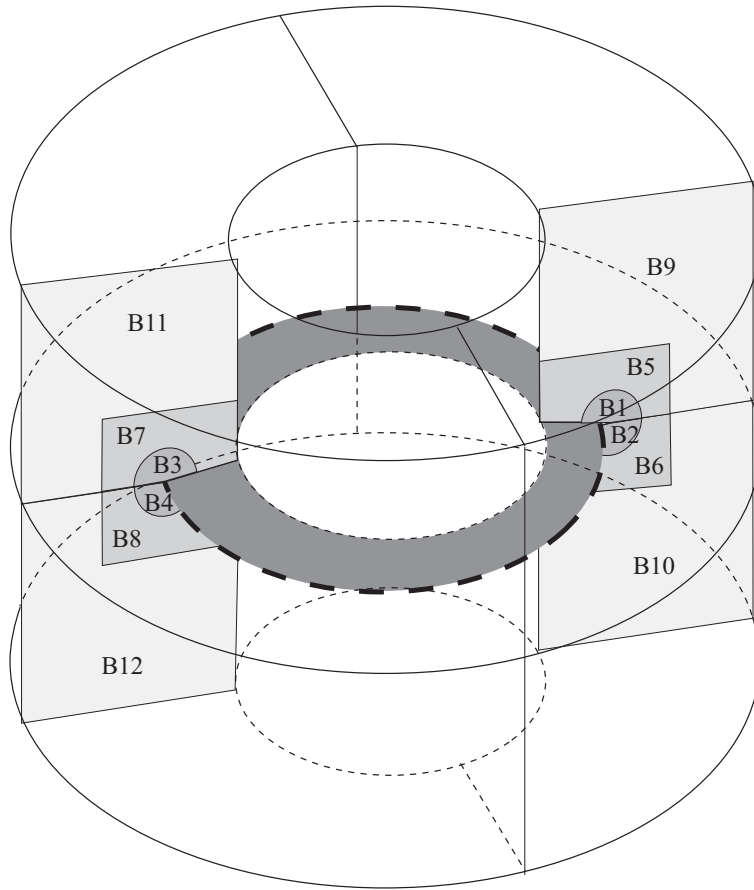
b) Crack front edge is a closed loop

Figure 10.13-1: Schematic division of domain for use of CRACK-M features, crack on boundary of model



a) Crack front edge is an open loop

Figure 10.13-2: Schematic division of domain for use of CRACK-M features, crack inside model (enclosed crack)



b) Crack front edge is a closed loop

Figure 10.13-2 (continued)

In all cases, the model is decomposed into three parts, the crack front, the crack sleeve and the remainder.

The crack front is that region of the model immediately adjacent to the crack front edge. This part of the model will be meshed using mapped meshing, as discussed below.

The crack sleeve is a transition region between the crack front and the remainder of the model. The crack sleeve encloses the crack front, and the crack sleeve should include the cracked surface not represented in the crack front itself. This part of the model will be meshed using free-form meshing.

The remainder of the model is constructed as usual, and is meshed using free-form meshing.

Each of these parts is represented with one or more geometry bodies, as shown. Due to the geometric modeler requirements, two bodies must be used to represent the closed loop crack front in Fig 10.13-1(b). Also, two bodies must be used to represent the enclosed crack front in Fig 10.13-2(a), and four bodies must be used to represent the enclosed closed loop crack front in Fig 10.13-2(b). For the remainder of the model, up to four bodies are used also, but these bodies could be merged into a single body, provided that the faces adjacent to the crack sleeve body remain compatible with the crack sleeve body faces.

Although a circular crack front is shown in the Figures, the CRACK-M features can be used whenever the geometry is topologically equivalent to the Figures.

Crack front body: A typical body used to represent the crack front is shown in Fig. 10.13-3. The body is prismatic, with one edge corresponding to the crack front edge, and one face corresponding to the cracked surface. Any geometric modeler tools can be used to create the crack front body, provided that the body is topologically equivalent to the body shown in Fig 10.13-3.

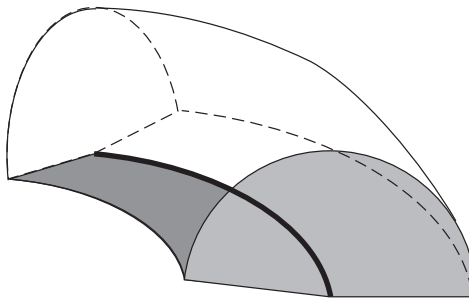


Figure 10.13-3: Body used to represent crack front

Crack sleeve body: Requirements on the crack sleeve body are: the crack sleeve body should completely surround the crack front body, the crack sleeve body should have one face on the cracked surface.

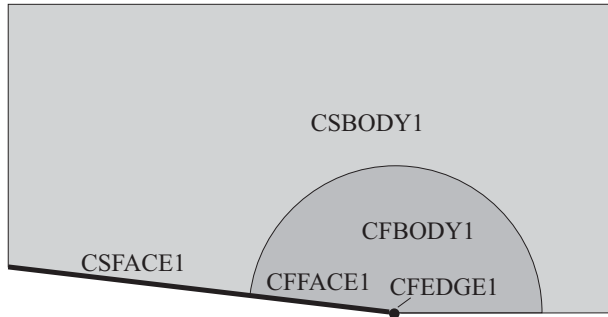
Again, any geometric modeler tools can be used to create the crack

sleeve body.

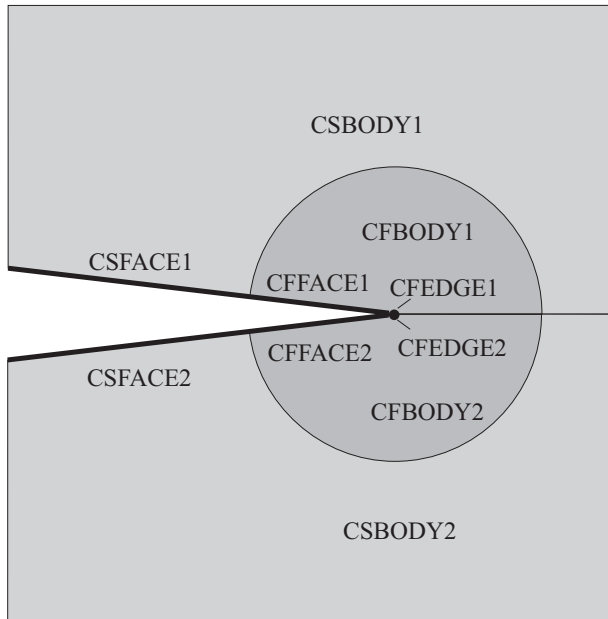
CRACK-M definition: Once the bodies have been defined, the CRACK-M DEFINE command is used to define the crack:

```
CRACK-M DEFINE NAME CFBODY1 CFFACE1 CFEDGE1,  
                   CFBODY2 CFFACE2 CFEDGE2,  
                   CSBODY1 CSFACE1,  
                   CSBODY2 CSFACE2
```

Fig 10-13.4 shows the parameters of the CRACK-M DEFINE command superposed on the geometry. Notice that when the crack is in the interior, the crack front edge is specified twice, once for each body used to represent the crack front.



a) Crack on boundary (see Fig 10.13-1)



b) Crack in interior (see Fig 10.13-2)

Figure 10.13-4: CRACK-M DEFINE examples

Although the cracked surface is shown at an angle in the figure, the cracked surface can be in the same plane as the uncracked ligament, and the two cracked surfaces for the crack in the interior

can be coincident.

When the crack front is a closed loop, two CRACK-M definitions must be made. Then the CRACK-M COMBINE command is used to combine the CRACK-Ms:

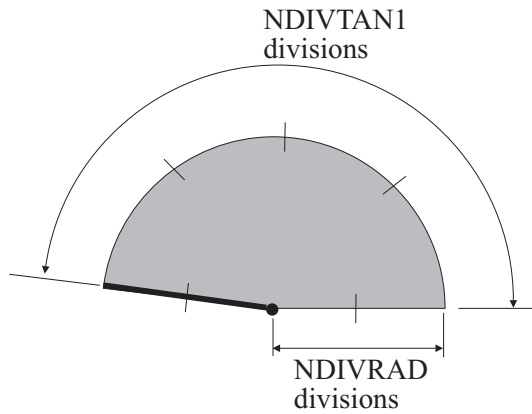
```
CRACK-M COMBINE NAME CRACKA CRACKB
```

where CRACKA, CRACKB are the two CRACK-Ms to be combined.

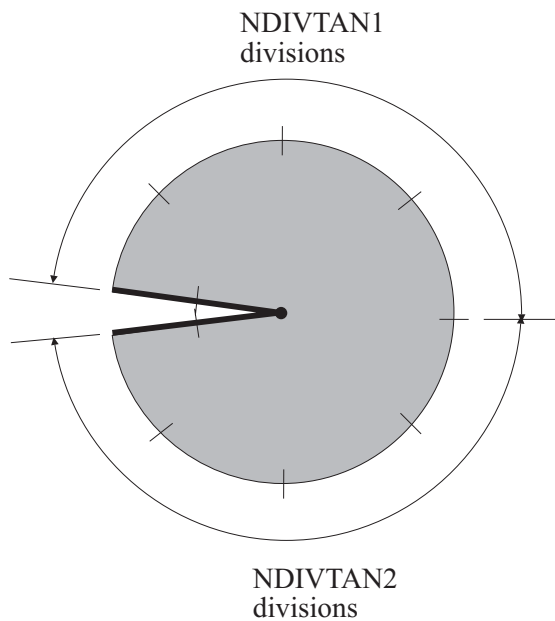
CRACK-M subdivision: The CRACK-M SUBDIVIDE command is used to subdivide the crack front bodies of the CRACK-M.

```
CRACK-M SUBDIVIDE NAME,  
NDIVCF NDIVTAN1 NDIVTAN2 NDIVRAD
```

NDIVCF is the number of subdivisions along the crack front edge. NDIVTAN1, NDIVTAN2 and NDIVRAD are the number of subdivisions in the other directions, see Fig 10.13-5. Notice that although the CRACK-M contains crack sleeve bodies, the crack sleeve bodies are not subdivided by CRACK-M SUBDIVIDE. The crack sleeve bodies are subdivided using the usual subdivision commands of the geometry modeler.



a) Crack on boundary (see Fig 10.13-1)



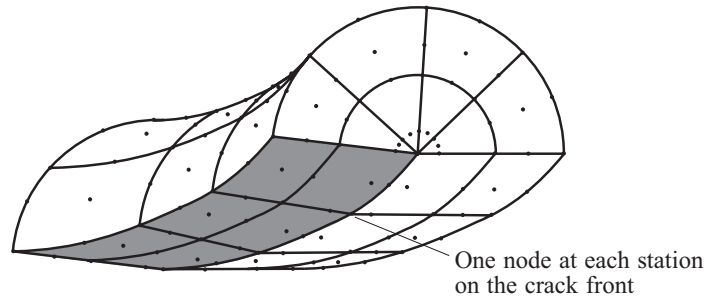
a) Crack in interior (see Fig 10.13-2)

Figure 10.13-5: CRACK-M SUBDIVIDE examples

Meshing: The CRACK-M MESH command is used to mesh the crack front bodies:

```
CRACK-M MESH NAME,  
      NODES NCTOLERANCE SUBSTRUCTURE GROUP  
      MIDNODES
```

where NAME is the name of a CRACK-M. Mapped meshing is used to generate a mesh with hexahedra and prisms adjacent to the crack front edge. The hexahedra can have either 8, 20 or 27-nodes. Fig. 10.13-6 shows typical meshes. At each station on the crack front edge, there is one node, and there is no option for generating multiple nodes at each station on the crack front edge. When the crack is in the interior of the model (so that the CRACK-M contains two crack surfaces), there is no coincidence checking on the crack surfaces, so separate nodes are generated at the matching positions of the crack surfaces.

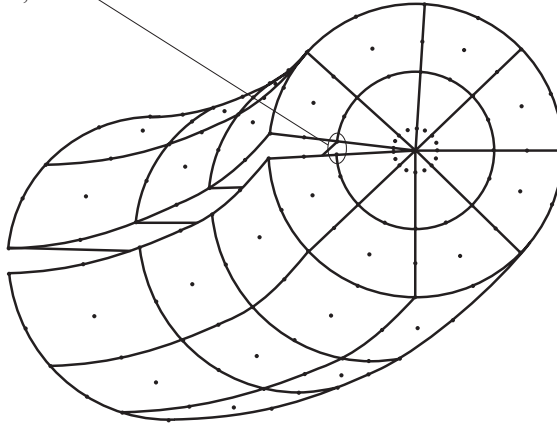


CRACK-M Q-POINT used to move nodes to the quarter-points

a) Crack on boundary

Figure 10.13-6: CRACK-M MESH examples

If cracked surfaces are coincident, separate coincident nodes are generated, for example, here:



b) Crack in interior of model

Figure 10.13-6: Continued

The CRACK-M Q-POINT command is used to move the mid-side and mid-face nodes of the (higher-order) elements adjacent to the crack front edges to the quarter-points.

CRACK-M Q-POINT POSITION

This command must be issued after the crack is meshed. When using the SVS method of virtual crack extension, it is not necessary to use the CRACK-M Q-POINT command, because the CRACK-SVS command can also shift the nodes to the quarter-points.

The CRACK-M GBODY command is used to mesh the crack sleeve bodies:

```
CRACK-M GBODY NAME,
      NODES NCTOLERANCE SUBSTRUCTURE GROUP
      MIDNODES
```

where NAME is the name of a crack sleeve body. Free-form meshing is used. When the CRACK-M contains two crack surfaces, there is no coincidence checking on the crack surfaces, so separate nodes are generated at the matching positions of the crack surfaces.

The remaining bodies are meshed using the usual geometry

modeler commands.

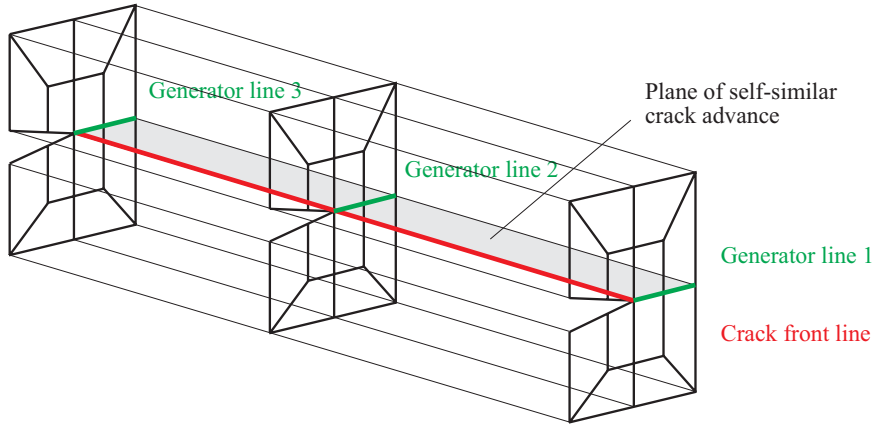
- It is strongly recommended that the FACELINK 1 OPTION=ALL command be used so that the meshing is compatible between adjacent bodies.
- When the crack is on the boundary of the model, it is necessary to assign proper boundary conditions to the uncracked ligament.

10.14 NVS method for virtual crack extension for 3-D cracks

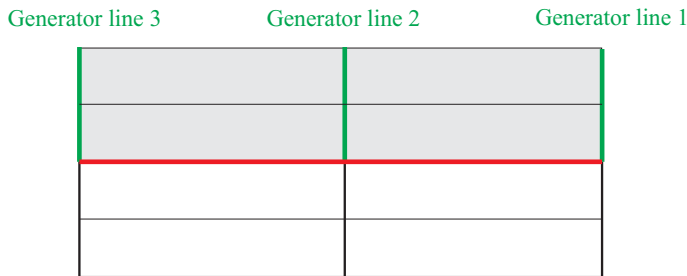
10.14.1 Overview

- In the NVS method for virtual crack extension, the virtual shift vector field is specified by virtual shift vectors at the nodal points.
- The fracture mechanics output consists of the values of the J-integral, calculated using the volume integral G (including corrections when applicable) for each of the virtual shifts. See Section 10.5 for a description of G .
- The NVS method makes use of a specific mesh layout near the crack front, as discussed below. If the model does not have this specific mesh layout near the crack front, the NVS method can still be used, but with limitations. Therefore, if the model does not have this specific mesh layout, the SVS method should be used instead.
- The following example shows nodal point virtual shift vectors for a 3-D mesh. Fig. 10.14-1 shows a 3-D mesh near the crack front. The mesh cross-section is the same for all generator lines. Element edges lie on the plane of self-similar crack advance. The plane of self-similar crack advance is also called the crack propagation surface.

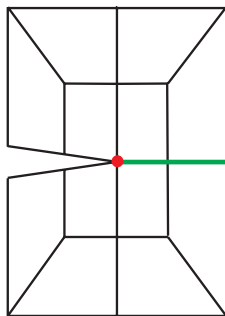
Although the crack front line is straight in the figure, the crack front line can also be curved.



a) Mesh layout near crack front, showing plane of self-similar crack advance, crack front line and three generator lines



b) Top view of crack propagation surface and open crack face



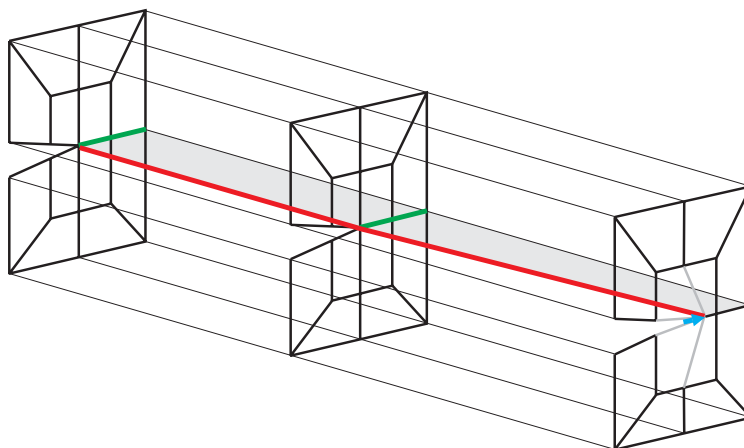
c) Cross-section at each generator line

Figure 10.14-1: Example mesh for description of NVS virtual shifts

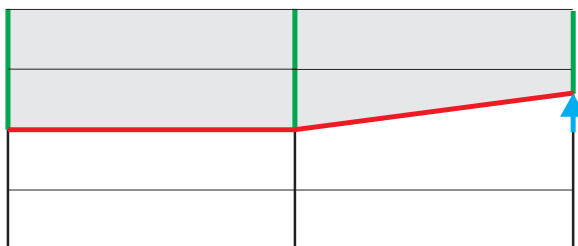
Fig. 10.14-2 shows a virtual shift for a crack advance at generator line 1. This virtual shift models a crack advance at the crack front node at generator line 1, with no crack advance at the other nodes on the crack front.

Fig. 10.14-3 shows a virtual shift applied at the crack front node and in one ring of elements at generator line 1. The same crack advance as in Fig 10.14-2 is modeled, however the region in which the virtual shift field is non-zero is larger than in Fig 10.14-2.

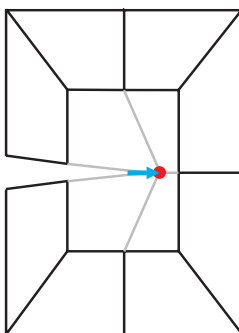
Figs. 10.14-4 and 10.14-5 show a virtual shift for a crack advance at generator line 2, both with the virtual shift applied at only one node, and also on one ring of elements.



a) Isometric view

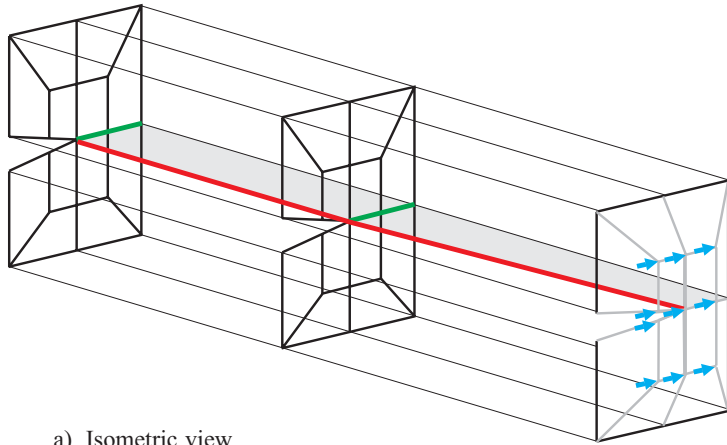


b) Top view of crack propagation surface and open crack face

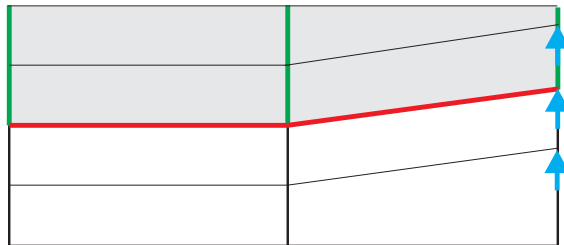


c) Cross-section at generator line 1

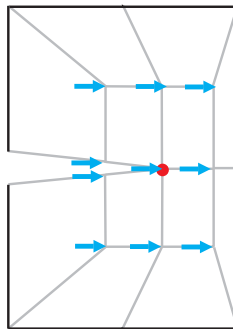
Figure 10.14-2: Virtual shift for crack front node at generator line 1, virtual shift applied at crack front only



a) Isometric view

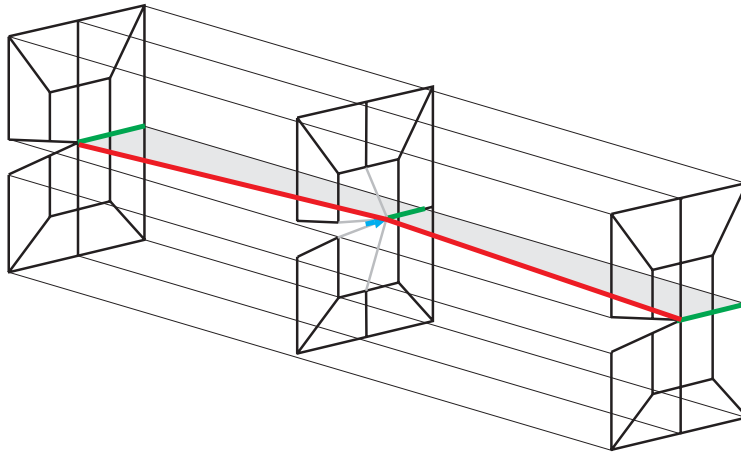


b) Top view of crack propagation surface and open crack face

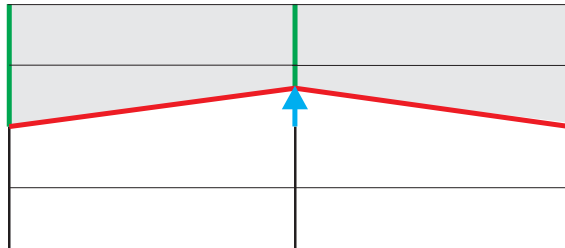


c) Cross-section at generator line 1

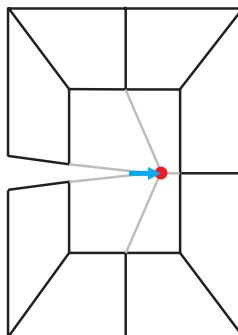
Figure 10.14-3: Virtual shift for crack front node at generator line 1, one ring of elements



a) Isometric view

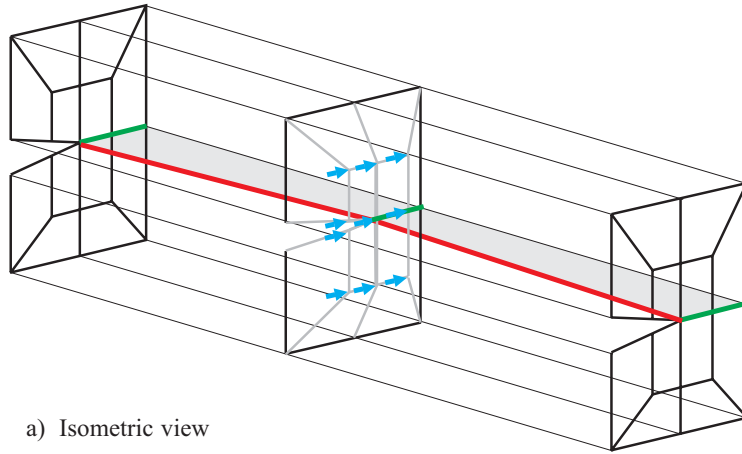


b) Top view of crack propagation surface and open crack face

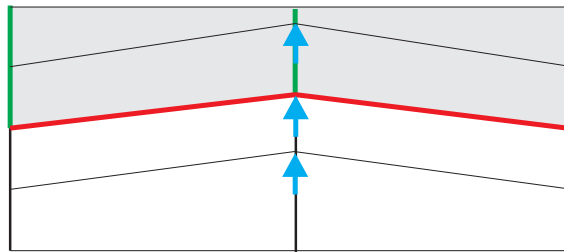


c) Cross-section at generator line 2

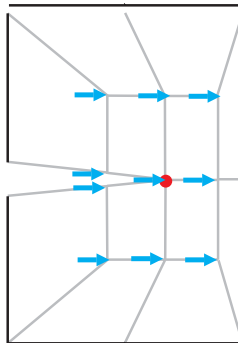
Figure 10.14-4: Virtual shift for crack front node at generator line 2, virtual shift applied at crack front only



a) Isometric view



b) Top view of crack propagation surface and open crack face



c) Cross-section at generator line 2

Figure 10.14-5: Virtual shift for crack front node at generator line 2, one ring of elements

- This example shows the general rule for specifying the virtual shift vectors. The virtual shift vector at the node of the crack front is specified. This virtual shift vector is applied to the node on the crack front, and also to a list of neighboring corner nodes on the cross-section at the given node. As will be seen later on, there are several ways in which this information is specified.
- This example also shows that the ideal mesh layout for use with the NVS method is as shown in Fig 10.14-6. The mesh perpendicular to the crack front is topologically “swept” along the crack front. This kind of meshing automatically insures that there is a mapped mesh of nodes both on the plane of self-similar crack advance and also along the open faces of the crack. Notice that there is a mesh cross-section corresponding to each generator line.

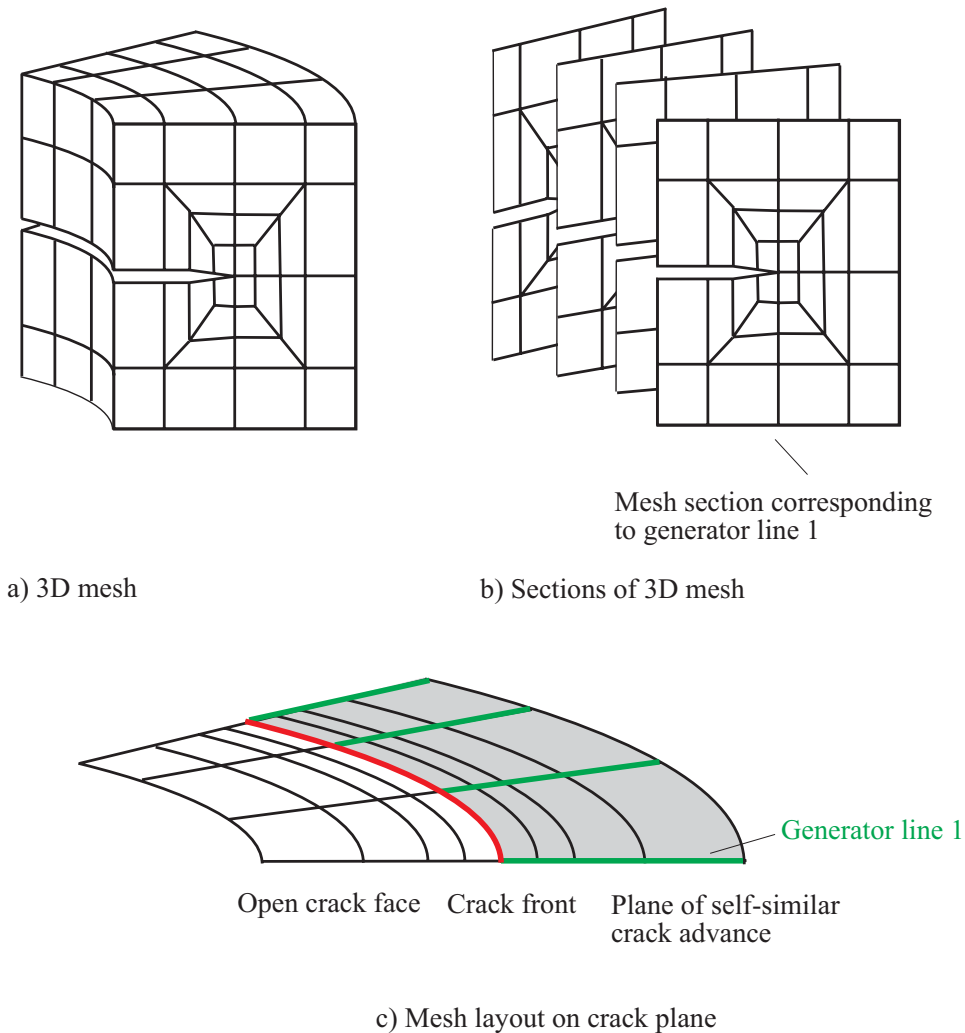


Figure 10.14-6: Ideal mesh layout for 3-D fracture model

- Corner nodes in the virtual shift are shifted by the full value of the virtual shift vector. Mid-side nodes are shifted by vectors interpolated from the corresponding corner nodes, so that a mid-side node between two corner nodes in the virtual shift is shifted by the full value of the virtual shift vector, and a mid-side node halfway between a corner node in the virtual shift and an unshifted corner node is shifted by half the value of the virtual shift vector.

Fig 10.14-7 gives an illustration.

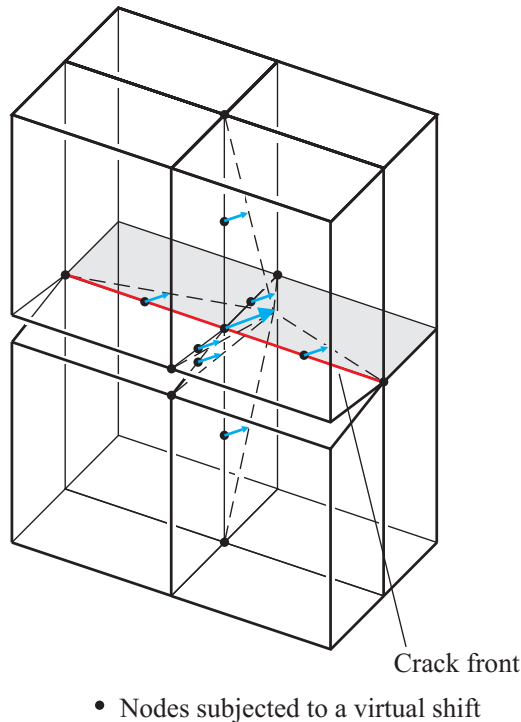


Figure 10.14-7: Virtual shifts at mid-side nodes in 3-D model

- As many virtual shifts as desired can be specified. There should be at least one virtual shift for each node on the crack front. Ideally there are several virtual shifts for each node on the crack front, each virtual shift being applied to more and more neighboring nodes in the element plane perpendicular to the crack front.
- If the mesh was created using the CRACK-M features, then the mesh already contains a mapped mesh of nodes on the plane of self-similar crack advance.

10.14.2 Definition of the crack propagation surface, CRACK-M features are not used

- The basic procedure for specifying the virtual shifts is

- ▶ Define a crack propagation surface using the CRACK-PROPAGATION commands
- ▶ Define the virtual shift vector and domains using the J-VIRTUAL-SHIFT commands.
- When there is no mapped mesh of nodes on the plane of self-similar crack advance, then only the crack front line can be specified. This specification is made using the AUI command

```
CRACK-PROPAGATION NODES  NCRACK=1  
node-1i
```

Here there is one data input line per node on the crack front.

- When there is a mapped mesh of nodes on the plane of self-similar crack advance, in which the mesh is topologically rectangular, then the crack front line and the plane of self-similar crack advance can be specified. There are several ways to make this specification:

By nodes: This specification is made using the AUI command

```
CRACK-PROPAGATION NODES  NCRACK>1  
node-1i node-2i ... node-NCRACKi
```

Here there is one data input line per generator line, with the first data input line corresponding to the first generator line, the second data input line corresponding to the second generator line, etc.

node-1_i is the node on the crack front for generator line *i*,
node-2_i is the next corner node along the generator line, etc.
Generator lines for mid-side and mid-face nodes should not be specified, that is, only corner nodes define generator lines.

By surfaces: This specification is made using the AUI command

```
CRACK-PROPAGATION SURFACE  
surfacei front-linei
```

Here there is one data input line per geometry line on the crack front.

- When only one side of the crack is modeled, ADINA automatically generates a zero prescribed displacement in the z direction for the crack front node and for each node on the crack propagation surface. If skew systems are used for the nodes on the crack propagation surface, the zero prescribed displacement is in the skew c direction.

10.14.3 Definition of the virtual shifts, CRACK-M features are not used

By node numbers: The most basic way to specify a virtual shift is to specify the virtual shift vector, and to specify the corner node numbers for the nodes shifted by the virtual shift vector.

The AUI command is

```
J-VIRTUAL-SHIFT NODE,  
                VECTOR VX VY VZ N3DSH  
nodei
```

Parameters VECTOR, VX, VY, VZ, N3DSH of the J-VIRTUAL-SHIFT NODE command are used to define the virtual shift vector, as follows:

VECTOR=INPUT: The shift vector is defined directly using parameters VX, VY, VZ. Note that both the magnitude and direction of this vector are used. This vector should lie in the plane of self-similar crack advance.

VECTOR=AUTOMATIC: The shift vector is defined automatically, as follows: The virtual shift vector has direction given by generator line N3DSH, and has magnitude of 1/100 the distance between the crack front node and the next corner node along this generator line.

The J-VIRTUAL-SHIFT NODE command defines only a single virtual shift at a time.

By rings: The virtual shifts can be defined in terms of rings.

The AUI command is

```
J-VIRTUAL-SHIFT RING,
```

```
VECTOR VX VY VZ,  
RING-TYPE=AUTOMATIC,  
RING-NUMBER,  
NORMAL={AUTOMATIC / INPUT} NX NY NZ,  
THICKNESS
```

Although this command has a number of options, we will discuss only the options RING-TYPE=AUTOMATIC and NORMAL=AUTOMATIC or INPUT, since the other options are impractical for 3-D cracks.

When RING-TYPE=AUTOMATIC, the AUI creates several virtual shift per corner node on the crack front, as follows. Suppose that there are NWCPs corner nodes on the crack front. Then

```
Virtual shift 1: crack front node at generator line 1  
Virtual shift 2: crack front node at generator line 2  
...  
Virtual shift NWCPs: crack front node at generator line  
NWCPs  
Virtual shift NWCPs+1: crack front node at generator line 1 +  
nodes from first ring of elements surrounding the crack front  
...  
Virtual shift 2*NWCPs: crack front node at generator line  
NWCPs + nodes from first ring of elements surrounding the  
crack front  
...
```

The total number of virtual shifts is NWCPs * (RING-NUMBER + 1). If RING-NUMBER = 0, then the virtual shifts are defined using only the crack front nodes. If RING-NUMBER = 1, then the virtual shifts are defined using the crack front nodes, and also the first ring of elements surrounding the crack front nodes.

The options for defining the virtual shift vectors are the same as for J-VIRTUAL-SHIFT NODE. When VECTOR=AUTOMATIC, the shift vector is determined for each crack front node using the generator line of that crack front node. When VECTOR=INPUT, parameters VX, VY, VZ are used to specify the virtual shift vector. Clearly VECTOR=INPUT is useful only for straight cracks.

Selection of nodes in the virtual shift: We now discuss how nodes are selected from the rings of elements surrounding the crack front. Consider RING-NUMBER=1. First, all of the nodes in the first ring of elements surrounding the entire crack front are selected. Then these nodes are tested against a disk, and only the nodes that lie within the given disk are selected.

The disk is defined by a node on the central plane, the orientation of the central plane, its radius and its thickness, as shown in Fig. 10.14-8. This information is specified as follows:

Node on the central plane. This is the corner node on the crack front, corresponding to the virtual shift to be defined. Thus each corner node on the crack front has its own disk.

Orientation of the central plane. This is determined by a normal vector as shown in the figure. This normal vector is specified using one of two options, as follows:

NORMAL=INPUT, normal vector is specified using parameters NX, NY, NZ.

NORMAL=AUTOMATIC, normal vector is perpendicular to the generator line of the corner node on the crack front, as shown in Fig 10.14-9.

Thickness of the disk: This is specified by parameter THICKNESS.

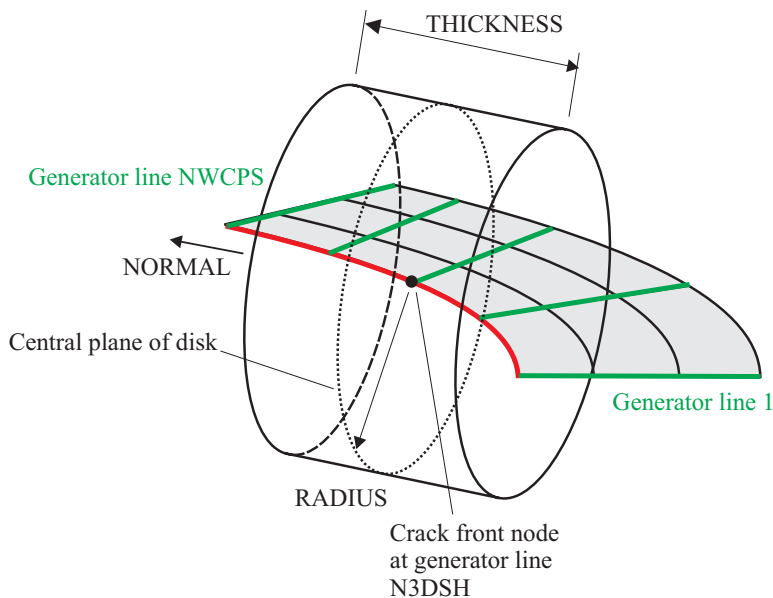
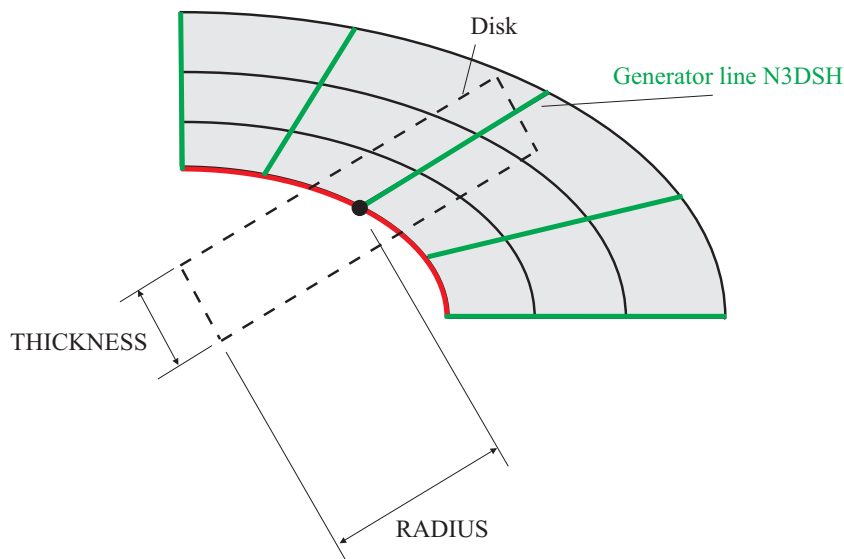


Figure 10.14-8: Definition of a disk used to select nodes in 3-D virtual shift



Crack propagation surface is viewed from above.

Figure 10.14-9: Disk defined using NORMAL=AUTOMATIC

- It is necessary to specify the crack propagation surface before using the J-VIRTUAL-SHIFT RING command.
- Although the J-VIRTUAL-SHIFT RING command accepts data input lines, data input lines should not be specified when using RING-TYPE=AUTOMATIC.

10.14.4 Definitions of the virtual shifts when CRACK-M features are used

When the CRACK-M features are used in the modeling of the crack, all of the virtual shifts are generated using the command

```
J-VIRTUAL-SHIFT CRACK-M NAME VSSCALE
```

where NAME is the number of the CRACK-M and VSSCALE is used to scale the length of the virtual shift vector.

The virtual shifts are generated in the following order:

Virtual shifts 1 to NDIVCF: Corner nodes at stations 1 to NDIVCF are shifted. Other corner nodes are not shifted.

Virtual shifts NDIVCF+1 to 2×NDIVCF: Corner nodes at stations 1 to NDIVCF are shifted. In addition, the first ring of nodes surrounding the corner nodes are shifted.

Virtual shifts 2×NDIVCF+1 to 3×NDIVCF: Corner nodes at stations 1 to NDIVCF are shifted. In addition, the first and second rings of nodes surrounding the corner nodes are shifted.

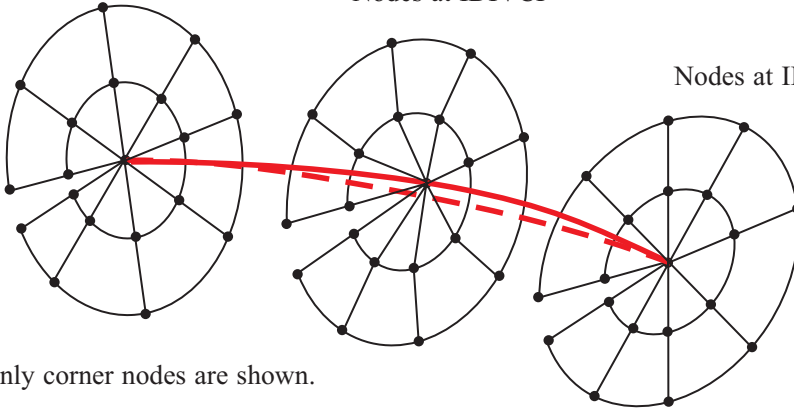
etc., up to virtual shift NDIVRAD×NDIVCF, in which all rings of nodes in the mapped mesh of the CRACK-M are shifted.

Fig. 10.14-10 shows typical virtual shifts for station IDIVCF. Fig 10.14-10(a) shows virtual shift IDIVCF and Fig 10.14-10(b) shows virtual shift NDIVCF+IDIVCF.

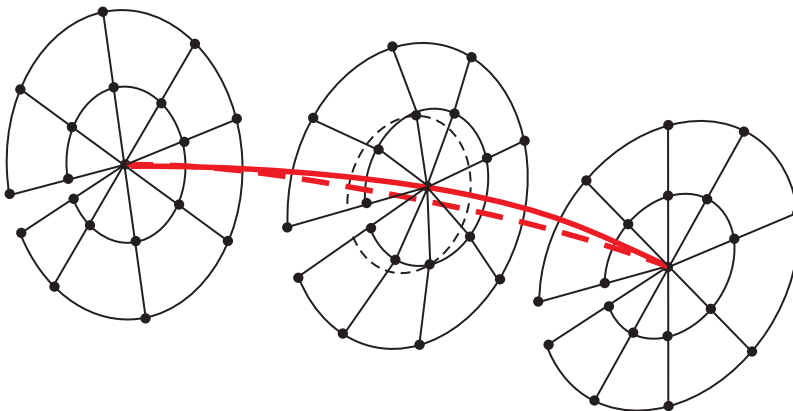
Nodes at IDIVCF+1

Nodes at IDIVCF

Nodes at IDIVCF-1



a) Virtual shift IDIVCF: Only node in ring 0 is shifted



b) Virtual shift NDIVCF + IDIVCF: Nodes in rings 0 and 1 are shifted

Figure 10.14-10: Virtual shifts for station IDIVCF

Each virtual shift vector for station IDIVCF is constructed as follows: The vector is normal to the geometric crack front edge at station IDIVCF, lies in the plane of the cracked surface, and points away from the cracked surface. The length of the vector is $VSSCALE \times (\text{radius of nodal ring } 1)$.

The command

```
LIST J-VIRTUAL-SHIFT CRACK-M
```

can be used to create a listing that gives, for each virtual shift, the crack front node, ring number, number of nodes and virtual shift vector. (The ring number is the number of nodal rings contained in the virtual shift.)

10.14.5 Fracture mechanics output

Virtual shift variables

Each virtual shift prints the following information, and outputs the following information to the porthole file. This information is accessed in the AUI using the given variable names.

J-PARAMETER_1:
(obsolete)

J-PARAMETER_2:
J-integral without thermal, pressure or dynamic corrections,
equal to G_I (G_I is the value of G from equation (10.5-1).

J-PARAMETER_3:
J-integral with temperature, pressure and dynamic
corrections, equal to $G_I + G_{th} + G_{pr} + G_{dyn}$.

TEMPERATURE_CORRECTION, PRESSURE_CORRECTION,
DYNAMIC_CORRECTION:

The corrections G_{th} , G_{pr} , G_{dyn}

Other variables

VIRTUAL_SHIFT_*_PART_FACTOR

The virtual shift participation factors at the nodes (for example, VIRTUAL_SHIFT_1_PART_FACTOR can be used to access the participation factors for virtual shift 1 at the nodes).

10.15 SVS method for virtual crack extension for 3-D cracks

10.15.1 Overview

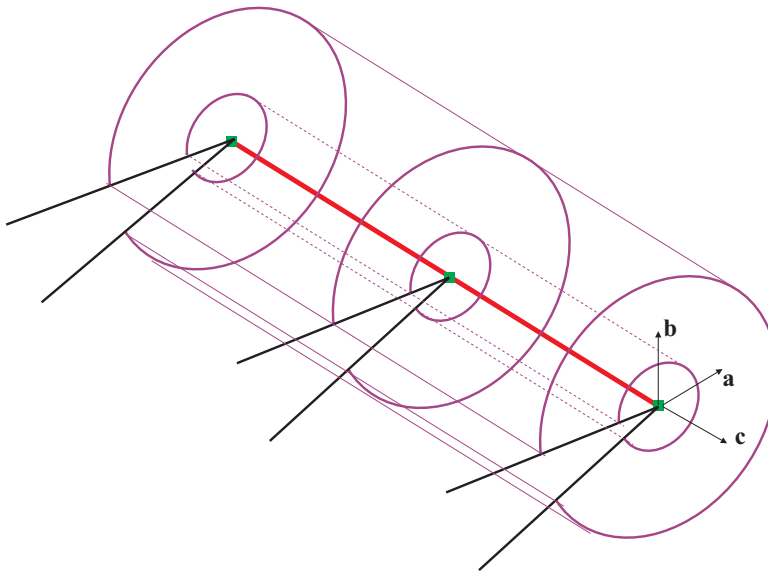
- In the SVS method for virtual crack extension, the virtual shifts are defined using crack advance stations, as described below.
- The fracture mechanics output consists of the values of the J-integral, calculated using the volume integral G (including corrections when applicable) for each of the virtual shifts. See Section 10.5 for a description of G . In addition, for a linear isotropic elastic analysis without thermal or dynamic effects, the fracture mechanics output also consists of the stress intensity factors K_I , K_{II} , K_{III} calculated for each of the virtual shifts. See Section 10.3 for a description of the stress intensity factors and Section 10.6 for a description of how the stress intensity factors are calculated.
- The SVS method does not rely on any specific mesh layout near the crack front. The mesh layout can be mapped or free-form. However, the crack front must be sharp, namely the crack front must be formed from the intersection of the crack top faces and the crack bottom faces.
- It is recommended that the mid-side nodes adjacent to the crack front be shifted to the quarter-points, in order to generate the $\frac{1}{\sqrt{r}}$ singularity near the crack front. This shifting is conveniently done as part of the definition of the SVS crack, see below.

10.15.2 Definitions of the virtual shifts

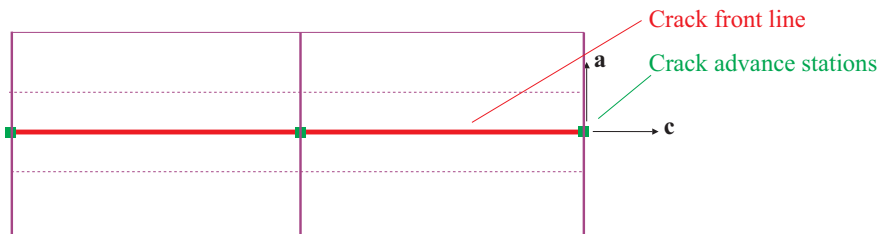
- The following example shows the virtual shift vector field near the crack front line. Fig. 10.15-1 shows a crack front and cracked faces near the crack front. In this example, the crack front line is straight, see below for an example in which the crack front line is curved. A coordinate system is aligned with the crack front line, as shown.

Crack advance stations are placed along the crack front line. These crack advance stations are not in general coincident with the node points.

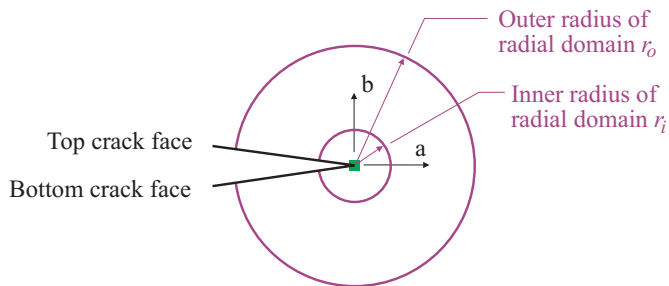
The crack front line is surrounded by a radial domain consisting of two parts, an inner radial domain and an outer radial domain. The radii of these domains are constant along the crack front.



a) Crack front, crack advance stations and radial domains



b) Top view of crack front

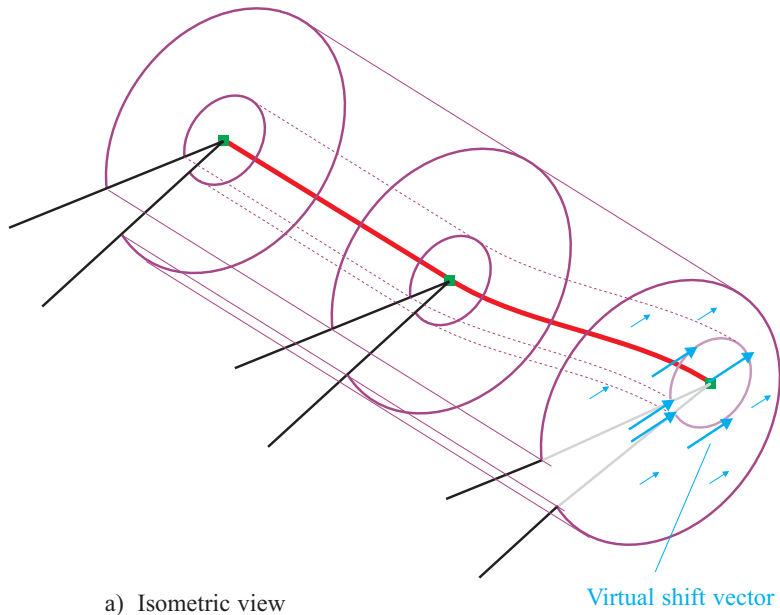


c) Side view of crack front, at crack advance station

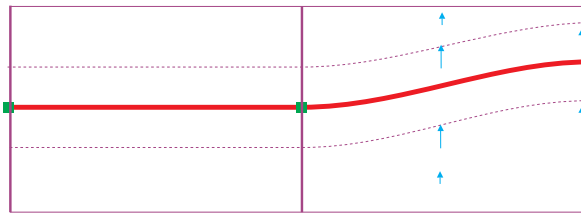
Figure 10.15-1: Crack front, crack advance stations, radial domains and coordinate system for SVS virtual shifts

Fig 10.15-2 shows how the crack advance is modeled for an advance of the end crack advance station. The virtual shift vector field $\Delta \mathbf{X}$ is shown in blue. This vector field is aligned with the local \mathbf{a} direction, and its magnitude varies within the domain both in the radial direction and in the axial direction. First, consider the vector field on the plane corresponding to the crack advance station (Fig 10.15-2(d)). On this plane, the vector field has full strength within the inner radial domain, and has zero strength outside the outer radial domain. Between the inner and outer radial domain, the magnitude of the vector field varies with the radius measured from the crack front. Next, consider the vector field on a plane halfway between the end crack advance station and the neighboring crack advance station. (Fig 10.15-2(c)). On this plane, the strength of the vector field is given by the strength of the vector field on the plane corresponding the crack advance station, multiplied by one-half. The strength of the vector fields on all other planes normal to the crack front is constructed similarly by interpolation. (The interpolations used will be discussed in more detail later.)

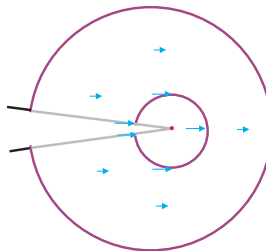
Fig 10.15-3 shows how the crack advance is modeled for an advance of an intermediate crack advance station. Again, this vector field is aligned with the local \mathbf{a} direction, and its magnitude varies within the domain both in the radial direction and in the axial direction. The same interpolation of the vector field is employed as already discussed for the end crack advance station.



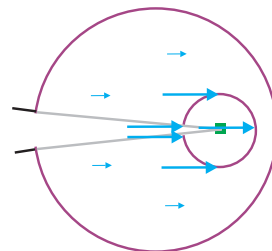
a) Isometric view



b) Top view

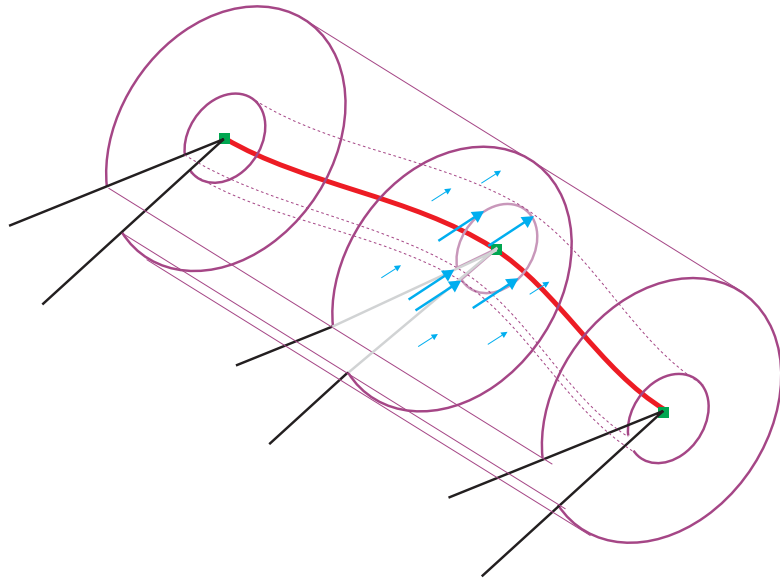


c) Side view of crack front, between crack advance stations

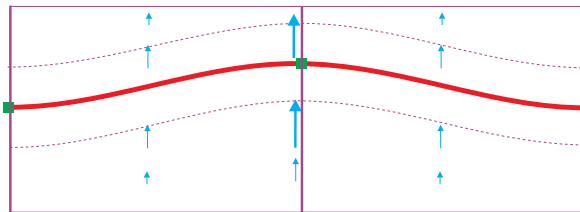


d) Side view of crack front, at crack advance station

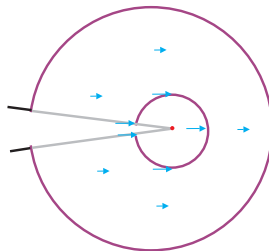
Figure 10.15-2: Modeling of crack advance at end crack advance station of open crack front



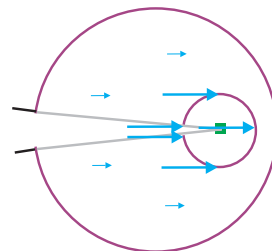
a) Isometric view



b) Top view



c) Side view of crack front, between crack advance stations



d) Side view of crack front, at crack advance station

Figure 10.15-3: Modeling of crack advance at intermediate crack advance station

- It is important to remember that the crack advance stations and radial domains are constructed without explicit reference to the finite element mesh surrounding the crack front. The virtual motion of the crack front is not represented in terms of nodal virtual shifts.
- Notice that the axial crack advance is not modeled using linear functions. Instead, cubic functions are used. The reasons for this choice are given below.
- When the crack front line is curved, very similar ideas are used. One difference concerns the construction of the local coordinate system. Fig. 10.15-4 shows a top view of the crack in the special case of a curved crack front modeled with straight element edges. The normal and tangential directions on the element edges are denoted \mathbf{n} and \mathbf{t} respectively. At the nodes, the \mathbf{n} directions are averaged to obtain average nodal point normals. The average nodal point normals are then interpolated to the crack advance station to obtain the local \mathbf{a} direction. Direction \mathbf{c} is then constructed to be orthogonal to direction \mathbf{a} and also to the crack plane direction \mathbf{b} .

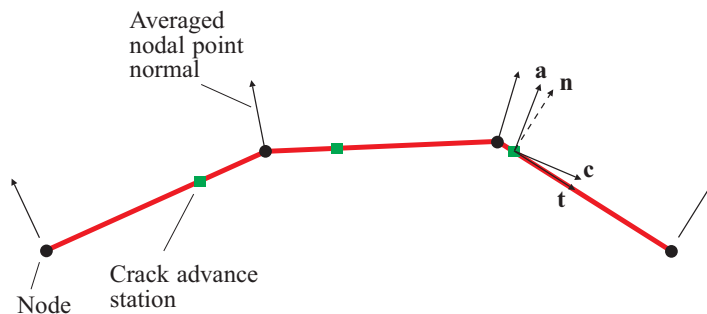


Figure 10.15-4: Crack front, crack advance stations and coordinate system for a curved crack front

In this way, the local coordinate directions \mathbf{a} , \mathbf{b} , \mathbf{c} vary continuously along the crack front.

The nodal point normal averaging is also done when the element edges are not straight.

Of course, the difference between the averaged and unaveraged normals can be made smaller by refining the mesh (assuming that the crack front to be modeled is continuous, and that the

discontinuities in the crack front are due only to mesh discretization).

- The local coordinates (r, θ, s) for global coordinate \mathbf{x} are determined as follows (Fig. 10.15-5). First the plane normal to the crack front that contains \mathbf{x} is found. This plane is associated with the distance along the crack front s . Then the projections of \mathbf{x} along local directions \mathbf{a} and \mathbf{b} give local coordinates a and b . The polar coordinates (r, θ) are easily obtained from (a, b) .

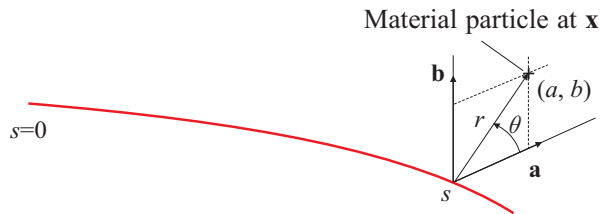


Figure 10.15-5: Local coordinates for a material particle

If there is more than one plane normal to the crack front that contains \mathbf{x} , the plane for which r is smallest is used. This situation can occur when the crack front is a closed loop.

- The outer radius of the radial domain of integration must be smaller than the radius of curvature of the crack front. Otherwise the local coordinates cannot be uniquely determined within the radial domain of integration, and the radial domain is degenerate (Fig. 10.15-6).

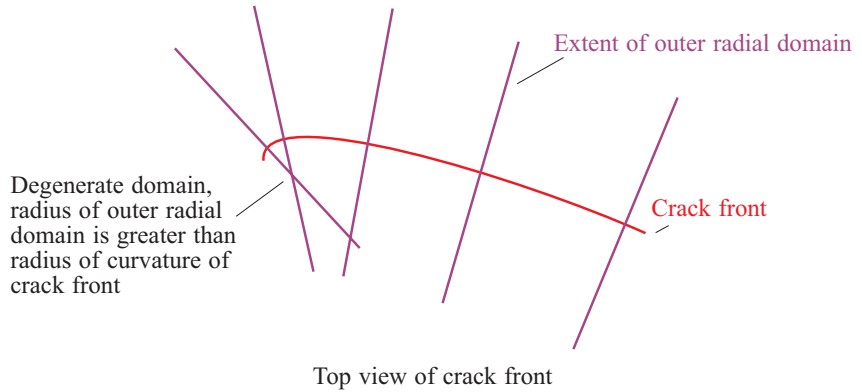


Figure 10.15-6: Degenerate radial domain

- In terms of the local coordinates defined above, the virtual shift field evaluated at position \mathbf{x} for a unit crack advance at station ICAS is given by

$$\Delta \mathbf{X}|_{\text{ICAS}} = q_r q_s|_{\text{ICAS}} \mathbf{a} \quad (10.15-1)$$

In this equation, direction \mathbf{a} is considered a function of the crack front coordinate s . The radial decay of the virtual shift field is given by the cubic function shown in Fig. 10.15-7(a):

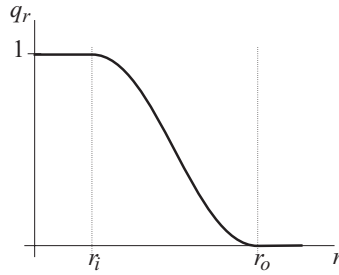
$$\begin{aligned} q_r &= 1, & r < r_i \\ &= 1 - 3u_r^2 + 2u_r^3, & r_i \leq r < r_o \\ &= 0, & r \geq r_o \end{aligned} \quad (10.15-2)$$

in which $u_r = \frac{r - r_i}{r_o - r_i}$. The decay of the virtual shift field along the crack front is given by the cubic functions shown in Fig. 10.15-7(b):

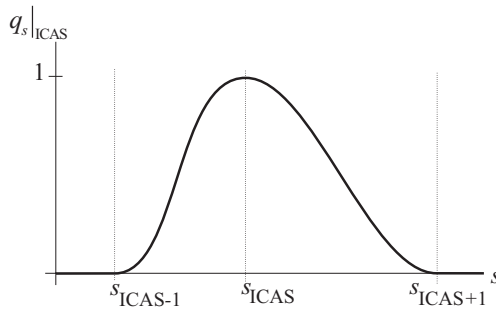
$$\begin{aligned}
 q_s|_{\text{ICAS}} &= 0, & s < s_{\text{ICAS}-1} \text{ or } s > s_{\text{ICAS}+1} \\
 &= 3u_s^2|_{\text{ICAS}-1} - 2u_s^3|_{\text{ICAS}-1}, & s_{\text{ICAS}-1} \leq s < s_{\text{ICAS}} \\
 &= 1 - 3u_s^2|_{\text{ICAS}} + 2u_s^3|_{\text{ICAS}}, & s_{\text{ICAS}} \leq s < s_{\text{ICAS}+1}
 \end{aligned} \tag{10.15-3}$$

in which $s_{\text{ICAS}-1}, s_{\text{ICAS}}, s_{\text{ICAS}+1}$ are the distances along the crack front at the successive stations ICAS-1, ICAS, ICAS+1, and in

$$\text{which } u_s|_{\text{ICAS}} = \frac{s - s_{\text{ICAS}}}{s_{\text{ICAS}+1} - s_{\text{ICAS}}}.$$



a) Radial decay



b) Axial decay

Figure 10.15-7: Cubic interpolation functions for radial and axial decay

- The gradient of the virtual shift field is obtained in two stages. First the gradient is obtained in the local coordinate system:

$\frac{\partial \Delta \mathbf{X}}{\partial r}, \frac{\partial \Delta \mathbf{X}}{\partial \theta}, \frac{\partial \Delta \mathbf{X}}{\partial s}$, then the gradient is transformed into the global coordinate system: $\frac{\partial \Delta \mathbf{X}}{\partial x_i} = \left(\frac{\partial \Delta \mathbf{X}}{\partial x}, \frac{\partial \Delta \mathbf{X}}{\partial y}, \frac{\partial \Delta \mathbf{X}}{\partial z} \right)$.

- The virtual crack area for a crack advance at station ICAS is given by equation (10.5-2), with the use of equation (10.15-1):

$$\Delta A_c|_{\text{ICAS}} = \int (\mathbf{a} \cdot \mathbf{n}) q_s|_{\text{ICAS}} ds \quad (10.15-4)$$

10.15.3 Calculation of J-integral and stress intensity factors

The following information is calculated for each virtual shift:

- The J-integral of the finite element solution (equation (10.5-1) along with the corrections (10.5-5 to 10.5-7)).

For static analysis in which the elements are all linear elastic isotropic without thermal effects, the following additional information is calculated:

- The interaction integrals between the finite element solution and the fundamental fields (e.g. $G(\mathbf{u}, \mathbf{u}^I)$), and also the interaction integrals of the fundamental fields (e.g. $G(\mathbf{u}^I)$). Equation (10.6-2) and the correction equation (10.6-13) are used to calculate G .
- The stress intensity factors K_I, K_{II}, K_{III} . The matrix equation (10.6-16) is used, with the numerically calculated values of the interaction integrals of the fundamental fields used on the left-hand-side of that equation.

The stress intensity factors K_I, K_{II}, K_{III} are calculated for boundary cracks, even for problems in which K_{II}, K_{III} are expected to be zero due to symmetry.

Details of the numerical integrations used are given in Section 10.15.4.

- One exception concerns the calculations of stress intensity factors at the ends of an open crack front. Notice that the axial decay q_s has a net change in value along the domain used at the end of an open crack front (Fig 10.15-2), therefore the integral $\int \frac{\partial q_s}{\partial s} ds$ is not equal to zero. As a consequence, the second term in equation (10.6-10) is non-zero, in other words, the fundamental fields for modes II and III are not orthogonal within the domain. Therefore ADINA computes only the stress intensity factor K_I for virtual shifts at the ends of an open crack front.

10.15.4 Calculation of G using numerical integration

- All integrals involved in the calculation of G are evaluated numerically, using the integration points and weights of the underlying finite element mesh. Thus, for example, the integral

$$\int_V \left(\tau_{ij} \frac{\partial u_j}{\partial x_k} - W \delta_{ik} \right) \frac{\partial \Delta X_k}{\partial x_i} dV$$

is evaluated using numerical

integration of the integrand $\left(\tau_{ij} \frac{\partial u_j}{\partial x_k} - W \delta_{ik} \right) \frac{\partial \Delta X_k}{\partial x_i}$. In this

integrand, τ_{ij} , $\frac{\partial u_j}{\partial x_k}$, W are obtained from the finite element

solution at the integration point, and $\frac{\partial \Delta X_k}{\partial x_i}$ is obtained from the

formulas given above. We will discuss the accuracy of this integration below.

- For calculation of stress intensity factors, it is also required to calculate expressions such as $G(\mathbf{u}')$ and $G(\mathbf{u}, \mathbf{u}')$. In this calculation, the fundamental solutions in Section 10.3 are used, assuming plane strain conditions, and using the local coordinates (r, θ) .
- The accuracy of the calculated stress intensity factors depends on

- The accuracy of the numerical integrations used in G
- The accuracy of the finite element solution displacement field \mathbf{u} .

These two accuracies are separate from each other. It is possible for the finite element solution field to be accurate and the numerical integrations used in G to be inaccurate, and vice versa.

- In this section, we discuss the accuracy of the numerical integrations used in G , without consideration of the accuracy of the finite element solution field.
- The accuracy of the numerical integrations depends upon the finite element mesh employed. For example, consider the integral

$$V|_{\text{ICAS}} = \int_{V|_{\text{ICAS}}} dV \quad (10.15-5)$$

in which $V|_{\text{ICAS}}$ corresponds to the domain in which $\Delta \mathbf{X}|_{\text{ICAS}}$ is non-zero (Fig 10.15-8).

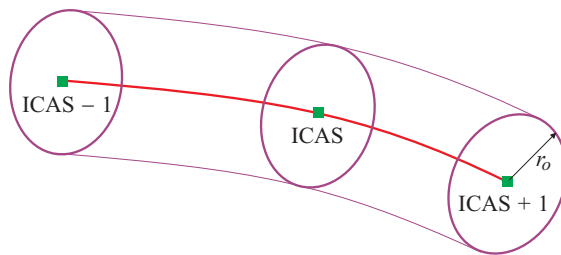


Figure 10.15-8: Domain over which $\Delta \mathbf{X}|_{\text{ICAS}}$ is non-zero

The analytical value of this integral is

$$V|_{\text{ICAS}} = \pi r_o^2 (s_{\text{ICAS}+1} - s_{\text{ICAS}-1}) \quad (10.15-6)$$

If a finite element mesh is used in which this domain is completely filled by elements, and in which the element boundaries correspond to the domain boundaries, then this integral is exactly calculated by the usual numerical integration used in finite element analysis. But for an arbitrary mesh, the domain contains partial elements (that is, elements cut by the domain boundary), and the numerical integration is not exact.

On the other hand, as the mesh is refined, keeping the stations and radial domains constant, we expect that the size of all of the partial elements will become smaller, so that the numerical integration becomes more accurate.

- To get an idea of the rate of convergence as the mesh is refined, consider a cube meshed with cube elements, and in each cube element, consider a cubic lattice of integration points each with uniform weight. Thus the entire domain contains a uniformly spaced lattice of integration points. If there are $N^{1/3}$ integration points in each direction, there are N integration points total.

Now consider the calculation of the volume of a domain that is just within the mesh of cube elements. This volume can be calculated by numerical integration using the uniformly spaced lattice of integration points. Any error in the integration will be confined to the boundary integration points. In this case, there are approximately $6N^{2/3}$ integration points on the boundary. In the worst case, these boundary integration points, that should have contributed to the numerical integration, do not contribute. The

relative error is then $\frac{6N^{2/3}}{N} = \frac{6}{N^{1/3}}$. Clearly as N increases, the error decreases, but the rate at which the error decreases is low.

Another way to express this rate of convergence is in terms of the element size h . Since the total volume is proportional to Nh^3 the relative error is proportional to h .

These results hold when integrating an arbitrary function over the domain, when the function has non-zero value on the boundary. These results also hold (with different constants of proportionality) for arbitrarily shaped domains with smooth boundary.

When the integrand has zero value on the boundary, but is proportional to h near the boundary, then the relative error in the integral becomes proportional to h^2 .

- Fortunately, for the integrals used in the calculation of G , it is possible to set the integrand equal to zero at the boundary. The integrand is set to zero at the boundary by proper choice of the virtual shift field $\Delta \mathbf{X}|_{\text{ICAS}}$, as follows.

The integrals that need to be evaluated have forms similar to $\int_V \left(\tau_{ij} \frac{\partial u_j}{\partial x_k} - W \delta_{ik} \right) \frac{\partial \Delta X_k}{\partial x_i} dV$. Therefore the integrals can be set to zero on the boundary by choosing $\Delta \mathbf{X}|_{\text{ICAS}}$ so that $\frac{\partial \Delta X_k}{\partial x_i}$ is zero on the boundary. Since $\Delta \mathbf{X}|_{\text{ICAS}} = q_r q_s|_{\text{ICAS}} \mathbf{a}$, $\frac{\partial \Delta X_k}{\partial x_i}$ can be set to zero on the boundary by setting both $q_r, \frac{\partial q_r}{\partial r}$ to zero on the boundary $r = r_o$, and by setting both $q_s|_{\text{ICAS}}, \frac{\partial q_s}{\partial s}|_{\text{ICAS}}$ to zero on the boundary $s = s_{\text{ICAS}-1}, s = s_{\text{ICAS}+1}$.

These considerations suggest the use of the cubic interpolation functions for q_r and $q_s|_{\text{ICAS}}$ already presented above in equations (10.15-2) and (10.15-3). Clearly both q_r and $\frac{\partial q_r}{\partial r}$ are equal to zero on the boundary $r = r_o$, and both $q_s|_{\text{ICAS}}$ and $\frac{\partial q_s}{\partial s}|_{\text{ICAS}}$ are equal to zero on the boundaries $s = s_{\text{ICAS}-1}, s = s_{\text{ICAS}+1}$.

Considering the numerical integration of integrals such as $\int_V \left(\tau_{ij} \frac{\partial u_j}{\partial x_k} - W \delta_{ik} \right) \frac{\partial \Delta X_k}{\partial x_i} dV$, for elements entirely within the domain, it is recognized that using linear interpolations for $\Delta \mathbf{X}|_{\text{ICAS}}$ (as in the NVS method) corresponds to multiplying $\tau_{ij} \frac{\partial u_j}{\partial x_k} - W \delta_{ik}$ by constants. Therefore if the numerical integration

of $\tau_{ij} \frac{\partial u_j}{\partial x_k} - W \delta_{ik}$ is exact in an element, the numerical integration

of $\left(\tau_{ij} \frac{\partial u_j}{\partial x_k} - W \delta_{ik} \right) \frac{\partial \Delta X_k}{\partial x_i}$ (in which linear interpolations are used

for $\Delta \mathbf{X}|_{\text{ICAS}}$) is also exact. On the other hand, using cubic

interpolations for $\Delta \mathbf{X}|_{\text{ICAS}}$ corresponds to multiplying

$\tau_{ij} \frac{\partial u_j}{\partial x_k} - W \delta_{ik}$ by quadratic functions. So even if the numerical

integration of $\tau_{ij} \frac{\partial u_j}{\partial x_k} - W \delta_{ik}$ is exact in an element, the numerical

integration of $\left(\tau_{ij} \frac{\partial u_j}{\partial x_k} - W \delta_{ik} \right) \frac{\partial \Delta X_k}{\partial x_i}$ (in which cubic

interpolations are used for $\Delta \mathbf{X}|_{\text{ICAS}}$) is not exact. Thus we have

introduced some error into the numerical integration process

through the use of cubic interpolations for $\Delta \mathbf{X}|_{\text{ICAS}}$.

On the other hand, the rate of convergence for numerical integration, for elements fully contained in the domain, is proportional to the size of the element h raised to a power of at least 2. (For example, for rectangular Gauss integration, the rate of convergence is proportional to h^{2m} where m is the number of Gauss points in each direction.) This rate of convergence is faster than the original rate of convergence (proportional to h) obtained when the integrand is non-zero on the domain boundary. This rate of convergence is also at least as fast as the rate of convergence anticipated when the integrand is proportional to h near the boundary.

So we anticipate that the relative error of the numerical integration of the integrals used for the calculation of G is proportional to h^2 , where h is the element size.

- The above considerations are based on worst-case conditions. In practice, better rates of convergence are often observed.

- There are several ways to assess the actual accuracy of the numerical integrations performed in the calculation of G . One way is to numerically calculate an integral in which the actual value is known, and in which the behavior of that integral is similar to the behavior of the integrals used in G .

Close to the crack front, $\tau_{ij} \frac{\partial u_j}{\partial x_k} - W \delta_{ik}$ has magnitude

proportional to $1/r$. Therefore the integral

$$\int_V \left(\tau_{ij} \frac{\partial u_j}{\partial x_k} - W \delta_{ik} \right) \frac{\partial \Delta X_k|_{\text{ICAS}}}{\partial x_i} dV \text{ is similar to the integral}$$

$$\int_V \frac{1}{r} \frac{\partial (q_r q_s|_{\text{ICAS}})}{\partial r} dV.$$

The integral $\int_V \frac{1}{r} \frac{\partial (q_r q_s|_{\text{ICAS}})}{\partial r} dV$ can be evaluated analytically over the given domain as follows:

$$\begin{aligned} \int_V \frac{1}{r} \frac{\partial (q_r q_s|_{\text{ICAS}})}{\partial r} dV &= \int_{s_{k-1}}^{s_{k+1}} \int_0^{2\pi} \int_0^{r_o} \frac{1}{r} \frac{\partial (q_r q_s|_{\text{ICAS}})}{\partial r} r dr d\theta (\mathbf{a} \cdot \mathbf{n}) ds \\ &= 2\pi \int_{s_{k-1}}^{s_{k+1}} \int_0^{r_o} \frac{1}{r} \frac{\partial (q_r q_s|_{\text{ICAS}})}{\partial r} r dr (\mathbf{a} \cdot \mathbf{n}) ds \\ &= 2\pi \int_{s_{k-1}}^{s_{k+1}} \int_0^{r_o} \frac{\partial q_r}{\partial r} q_s|_{\text{ICAS}} dr (\mathbf{a} \cdot \mathbf{n}) ds \\ &= 2\pi \left(q_r|_{r_o} - q_r|_0 \right) \int_{s_{\text{ICAS}-1}}^{s_{\text{ICAS}+1}} q_s|_{\text{ICAS}} (\mathbf{a} \cdot \mathbf{n}) ds \\ &= -2\pi \Delta A_c|_{\text{ICAS}} \end{aligned}$$

So define

$$GTEST|_{\text{ICAS}} = -\frac{1}{2\pi \Delta A_c|_{\text{ICAS}}} \int_V \frac{1}{r} \frac{\partial (q_r q_s|_{\text{ICAS}})}{\partial r} dV \quad (10.15-7)$$

Based on the above analysis, we anticipate that this integral, when evaluated using the same numerical integration as used for the

evaluation of G , should converge to the exact value of 1 with a rate of convergence of h^2 when cubic interpolation functions are used for q_r and $q_s^{(k)}$. And the accuracy of *GTEST* should be similar to the accuracy of G .

For a boundary crack, substitute π for 2π in the above expressions.

- Another way to confirm the actual accuracy of the numerical integrations is to compare the calculated values of $G(\mathbf{u}^I)$, $G(\mathbf{u}^{II})$, $G(\mathbf{u}^{III})$, $G(\mathbf{u}^I, \mathbf{u}^{II})$, $G(\mathbf{u}^I, \mathbf{u}^{III})$, $G(\mathbf{u}^{II}, \mathbf{u}^{III})$ with their theoretical values (equations 10.6-6). This confirmation can be performed only when the crack is straight, however. Also recall that $G(\mathbf{u}^{II}, \mathbf{u}^{III})$ is theoretically non-zero for virtual shifts corresponding to the end crack advance stations of open crack fronts (equation 10.6-10).

10.15.5 Definitions of the cracks

Each crack is defined using the command

```
CRACK-SVS NAME,  
    ADV-OPTION ADV-NUMBER ADV-DISTANCE,  
    RADIUS-INNER RADIUS-MAXIMUM,  
    RADIUS-NUMBER RADIUS-RATIO,  
    BINORMAL-DIRECTION,  
    START-OPTION START-NODE START-POINT,  
    START-X START-Y START-Z START-SYSTEM,  
    SHIFT-OPTION SHIFT-VALUE GTYPE  
namei bodyi sidei
```

The parameters of this command are

NAME: number to assign to this crack. There can be more than one crack in the model.

ADV-OPTION, ADV-NUMBER, ADV-DISTANCE: Parameters used to place the crack advance stations along the crack front. The default is to place the crack advance stations at the corner nodes of

the element edges on the crack front.

RADIUS-INNER, *RADIUS-MAXIMUM*, *RADIUS-NUMBER*, *RADIUS-RATIO*: Parameters used to define the radial domains. The crack can have more than one radial domain. The default for *RADIUS-INNER* is 0, *RADIUS-NUMBER* is 1 and *RADIUS-RATIO* is 1. There is no default for *RADIUS-MAXIMUM*; *RADIUS-MAXIMUM* must be specified.

BINORMAL-DIRECTION: Parameter used to orient the crack binormal direction (crack plane direction) **b**. The default is to orient the binormal direction upwards.

START-OPTION, *START-NODE*, *START-POINT*, *START-X*, *START-Y*, *START-Z*, *START-SYSTEM*: Parameters used to choose the starting crack advance station for a closed crack front. The default is to set the starting crack advance station at the lowest corner node number.

SHIFT-OPTION, *SHIFT-VALUE*: Parameters used to shift the mid-side, mid-face and mid-volume nodes towards the crack front. The default is not to shift the mid-side, mid-face and mid-volume nodes of elements adjacent to the crack front to the quarter-points.

GTYPE: Parameter used to specify the type of input on the data input lines.

name_i, *body_i*, *side_i*: Data input lines used to select element faces on the crack faces.

If *GTYPE*=THREE-D, then *name_i* is either the number of a geometry surface (if (*body_i* = 0) or the number of a geometry face (if (*body_i* > 0). If *GTYPE*=ELFACSET, then *name_i* is the number of an element face-set and *body_i* is unused.

side_i specifies whether the specified geometry (or element face-set) corresponds to the top surface, bottom surface, cracked surface or uncracked ligament of the crack. For an enclosed crack, define the crack by specifying the geometry on the crack top and bottom surfaces. For a boundary crack, define the crack by specifying the geometry on the cracked surface and on the

uncracked ligament.

For full details, see the Command Reference Manual for ADINA, CRACK-SVS command.

The CRACK-SVS command determines the crack front edges as the element edges that are on both the top and bottom surface of the crack (enclosed crack), or the element edges that are on both the cracked surface and uncracked ligament (boundary crack).

The crack front edges can form an open curve (open crack front) or a closed curve (closed crack front). In the case when the crack front is a closed curve, the crack advance stations at the beginning and end of the curve are coincident.

Examples

Example 1:

```
ELFACESET 1 . . . // defines an element face-set
                  for the top surface of a crack
ELFACESET 2 . . . // defines an element face-set
                  for the bottom surface of the crack
CRACK-SVS RADIUS-MAXIMUM=2 GTYPE=ELFACESET
1 , , TOP
2 , , BOTTOM
DATAEND

// defines an SVS crack with number 1. The crack
// advance stations are at the corner nodes of the
// element edges on the crack front. Any mid-side
// nodes adjacent to the crack front are shifted to the
// quarter-points. There is one radial domain with inner
// radius zero and outer radius 2.
```

Example 2:

// Mesh is defined using geometry bodies. The top of the crack corresponds to faces 1 and 4 of body 2, and the bottom of the crack corresponds to faces 3 and 2 of body 4.

```
CRACK-SVS NAME=2 ADV-OPTION=NUMBER,
ADV-NUMBER=30,
```



```
RADIUS-INNER=0.1,
RADIUS-MAXIMUM=2,
RADIUS-NUMBER=5,
SHIFT-OPTION=NO

1 2 TOP
4 2 TOP
3 4 BOTTOM
2 4 BOTTOM
DATAEND

// defines an SVS crack with number 2. 30 crack
// advance stations are equally spaced around the crack
// front. There are five radial domains with maximum
// radius 2. Since the RADIUS-RATIO is 1.0, the
// spacing between successive radii is constant with
// value  $(2 - 0.1)/5 = 0.38$ . So the domains are
```

Domain 1: inner radius 0.1, outer radius 0.48 ($0.1 + 0.38$)
Domain 2: inner radius 0.1, outer radius 0.86 ($0.48 + 0.38$)
Domain 3: inner radius 0.1, outer radius 1.24
Domain 4: inner radius 0.1, outer radius 1.62
Domain 5: inner radius 0.1, outer radius 2.00

Any mid-side nodes adjacent to the crack front are not shifted.

10.15.6 Virtual shift numbers assigned to the virtual shifts

The CRACK-SVS command assigns a virtual shift number to each virtual shift.

In the following,

ICRACK = the label number of a crack defined with the CRACK-SVS command.

NCAS(ICRACK) = the number of crack advance stations for crack ICRACK.

NCRAD(ICRACK) = the number of radial domains for crack ICRACK.

NCVS(ICRACK) = the number of virtual shifts for crack

ICRACK.

Then the number of virtual shifts for crack ICRACK is given by

$$\text{NCVS(ICRACK)} = \text{NCAS(ICRACK)} * \text{NCRAD(ICRACK)}$$

For the first crack (crack with lowest label number), the virtual shifts are defined in the following order:

virtual shift 1 = virtual shift at crack advance station 1 for radial domain 1

virtual shift 2 = virtual shift at crack advance station 2 for radial domain 1

...

virtual shift NCAS = virtual shift at crack advance station NCAS for radial domain 1

virtual shift NCAS+1 = virtual shift at crack advance station 1 for radial domain 2

...

virtual shift 2*NCAS = virtual shift at crack advance station NCAS for radial domain 2

...

virtual shift NCRAD*NCAS = virtual shift at crack advance station NCAS for radial domain NCRAD.

When there is more than one crack defined in the model, then the virtual shift numbering for each successive crack starts with the highest virtual shift number for the preceding cracks + 1. For example, in a model with two cracks, the virtual shifts for the second crack are defined in the following order:

virtual shift NCVS(1) + 1 = virtual shift at crack advance station for radial domain 1

...

virtual shift NCVS(1) + NCRAD(2)*NCAS(2) = virtual shift at crack advance station NCAS(2) for radial domain NCRAD(2).

Example:

```
CRACK-SVS 1 . . . // defines an SVS crack with 7 crack  
                  advance stations and 3 radial domains
```

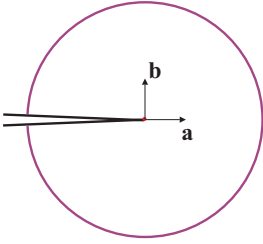
CRACK-SVS 5 . . . // defines an SVS crack with 4 crack
advance stations and 2 radial domains

The following virtual shift numbers are defined:

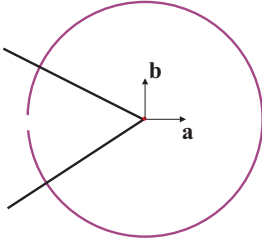
Virtual shift number	Crack-SVS number	Radial domain number	Advance station number
1	1	1	1
2	1	1	2
...			
7	1	1	7
8	1	2	1
...			
14	1	2	7
15	1	3	1
...			
21	1	3	7
22	5	1	1
...			
25	5	1	4
26	5	2	1
...			
29	5	2	4

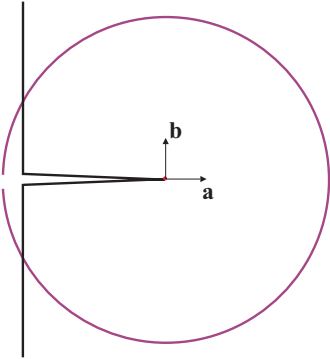
10.15.7 Modeling hints

- The crack faces must be flat and parallel to each other.
- Each radial domain must include at least one element integration point.
- Examples of good and bad modeling are shown in Figure 10.15-9.

- 

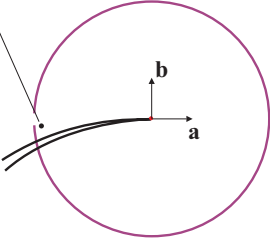
a) Crack faces are parallel and flat within the virtual shift domain, OK

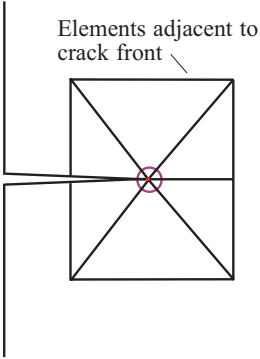


b) Crack faces are not parallel, not OK
- 

c) Radial domain lies outside of model domain, not OK

In the auxiliary field calculations, this point appears to lie on the bottom surface of the crack.



d) Crack faces are not flat within the radial domain, not OK
- 

e) Radial domain does not enclose any element integration points, not OK

Figure 10.15-9: Examples of good and bad modeling for SVS cracks

- It is allowed for the crack front to be curved. However, the theory used in the calculation of stress intensity factors assumes that the crack front is straight.

Therefore, in order that accurate results be obtained, the outer radius of the radial domain should be much smaller than the radius of curvature of the crack front. This may require a fine mesh near the crack front.

- If the mesh is unstructured, shifting the nodes to the quarter-points can lead to overdistorted meshes. This is why the default is not to shift the nodes to the quarter-points.
- The number of crack advance stations should not be greater than the number of corner nodes on the crack front.

10.15.8 Fracture mechanics output

VSINFO command

During post-processing, the VSINFO command can be used to output a number of useful tables that summarize the definitions of the virtual shifts, and which also output various error measures. The error measures are

```
SVS_VOLUME_PERCENT_ERROR  
SVS_GTEST_PERCENT_ERROR  
G-I_PERCENT_ERROR  
G-II_PERCENT_ERROR  
G-III_PERCENT_ERROR
```

(see descriptions of these error measures below).

Virtual shift variables

Each virtual shift prints the following information, and outputs the following information to the porthole file. This information is accessed in the AUI using the given variable names.

VIRTUAL_SHIFT_NUMBER:

The number of the virtual shift

SVS_CRACK_NUMBER:

The crack number

SVS_ADVANCE_STATION_NUMBER:

The crack advance station number

SVS_RADIAL_DOMAIN_NUMBER:

The crack radial domain number

VIRTUAL_SHIFT_DIRECTION- {XYZ} :

The direction of the virtual shift

SVS_CRACK_FRONT_LENGTH:

The total length of the crack front = $s_{NCAS} - s_1$

SVS_CRACK_FRONT_DISTANCE:

The distance along the crack front to the crack advance station = s_{ICAS}

SVS_LENGTH:

length of axial domain,

= $s_2 - s_1$ for a virtual shift at the start of an open crack front,

= $s_{NCAS} - s_{NCAS-1}$ for a virtual shift at the end of an open crack front,

= $s_{ICAS+1} - s_{ICAS-1}$ otherwise

SVS_INNER_RADIUS:

Inner radius of the radial domain r_i

SVS_OUTER_RADIUS:

Outer radius of the radial domain r_o

SVS_AREA:

Area of the virtual shift (equation 10.15-4)

SVS_ELEMENT_INT_POINTS:

Number of element integration points used in the domain.

SVS_PRESSURE_INT_POINTS:

Number of pressure load integration points used in the

domain.

SVS_VOLUME_CALCULATED:

Calculated volume of the domain, as computed using numerical integration

SVS_VOLUME_THEORETICAL:

Theoretical volume of the domain

SVS_VOLUME_PERCENT_ERROR:

Relative error in the volume calculation:

$$\text{SVS_VOLUME_PERCENT_ERROR} = 100 \times \left(\frac{\text{SVS_VOLUME_CALCULATED}}{\text{SVS_VOLUME_THEORETICAL}} - 1 \right)$$

SVS_PRESSURE_AREA_CALCULATED:

Calculated pressure area of the domain, as computed using numerical integration.

SVS_GTEST_CALCULATED:

Calculated value of the test integral $GTEST$, as computed using numerical integration. $GTEST$ is defined in equation (10.15-7).

SVS_GTEST_THEORETICAL:

Theoretical value of the test integral $GTEST$.

SVS_GTEST_PERCENT_ERROR:

Error in the calculation of $GTEST$:

$$\text{SVS_GTEST_PERCENT_ERROR} = 100 \times \left(\frac{\text{SVS_GTEST_CALCULATED}}{\text{SVS_GTEST_THEORETICAL}} - 1 \right)$$

G-FE:

The value of the J-integral computed from the finite element solution, as described in Section 10.5. For a crack in mode I loading, G_FE can be interpreted as the energy release rate. G_FE includes the pressure, temperature and dynamic corrections, if these corrections are requested in the FRACTURE command (the default is to request the corrections).

G-FE_UNCORRECTED:

The value of the J-integral computed from the finite element solution, without corrections.

G-FE_DYNAMIC_CORRECTION:

The value of the dynamic correction of the J-integral computed from the finite element solution.

G-FE_PRESSURE_CORRECTION:

The value of the pressure correction of the J-integral computed from the finite element solution.

G-FE_TEMPERATURE_CORRECTION:

The value of the temperature correction of the J-integral computed from the finite element solution.

G-FE-I, G-FE-II, G-FE-III:

The values of G as computed from the interaction of the finite element solution and the auxiliary mode I, II, III fields.

G-I, G-II, G-III:

The values of G as computed from the auxiliary mode I, II, III fields, calculated using numerical integration. For example $G-I = G(\mathbf{u}^I)$.

G-I-II, G-I-III, G-II-III:

The values of G as computed from the interactions of the auxiliary mode I, II, III fields, calculated using numerical integration. For example $G-I-II = G(\mathbf{u}^I, \mathbf{u}^{II})$. In theory, for a straight crack front, these values should all be zero (with the exception of G-II-III computed at the end crack advance stations of an open crack front).

G-I_THEORETICAL, G-II_THEORETICAL,

G-III_THEORETICAL:

The theoretical values of G as computed from the auxiliary mode I, II, III fields (assuming that the crack front is

straight). For example $G_I_THEORETICAL = \frac{1}{E'}$.

G-I_PERCENT_ERROR, G-II_PERCENT_ERROR,
G-III_PERCENT_ERROR:

The percent error in G as computed from the auxiliary mode I, II, III fields (assuming that the crack front is straight). For example

$$G-I_PERCENT_ERROR = 100 \times \left(\frac{G-I}{G-I_THEORETICAL} - 1 \right).$$

G-RESIDUAL:

G as computed from the residual field, namely the finite element solution minus the stress intensity factors times the auxiliary fields.

G-RESIDUAL_PERCENT:

The relative size of G-RESIDUAL, compared with G-FE, expressed as a percentage:

$$G-RESIDUAL_PERCENT = 100 \times \left(\frac{G-RESIDUAL}{G-FE} \right)$$

K-I, K-II, K-III:

The stress intensity factors. K-II, K-III are not computed for the end crack advance stations of an open crack front.

The above list assumes that the analysis is isotropic linear elastic without thermal or dynamic effects.

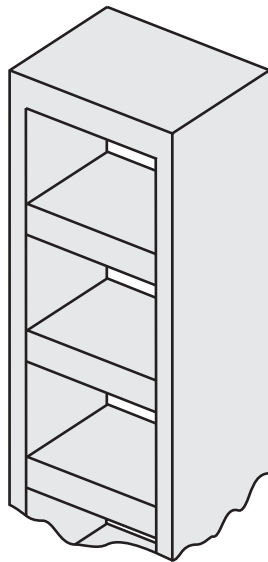
If the analysis is not isotropic linear elastic, or the analysis includes thermal or dynamic effects, the auxiliary field variables, residual field variables and stress intensity factors are not calculated.

11. Additional capabilities

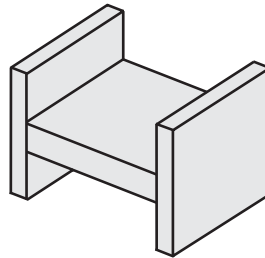
11.1 Substructuring

11.1.1 Substructuring in linear analysis

- As an alternative to assembling and directly solving the complete system in linear static analysis, substructuring can be employed. Through the use of substructuring, in essence, large new finite elements are established that are then assembled in the same way as finite elements. Figure 11.1-1 illustrates an application of substructuring. The computational effort saved through substructuring can be significant. Also, the time required for the preparation of the input data for a large problem may be significantly less when using substructuring.



Structure considered



Typical structure of one story
(consists of beam elements, plate
elements, membrane elements, etc...)

Figure 11.1-1: Substructuring in the analysis of a multistory building

- Each substructure can be reused one or more times; each use of the substructure is called a "reuse". In Figure 11.1-1, one substructure is defined, corresponding to one story of the building. The substructure is reused for each story.
- The nodes of a substructure are grouped into two categories, condensed and retained. A condensed node is one that is not connected to any nodes in the main structure, a retained node is connected to at least one node in the main structure.
- Substructuring in dynamic analysis can be employed when the trapezoidal rule of time integration is used in a direct step-by-step solution (see Section 7.3).
- Substructuring can be used in two different ways:

ref. KJB
Section 8.2.4

- ▶ If the mass is lumped to a relatively few degrees of freedom, the massless degrees of freedom can be statically condensed out by substructuring. Figure 11.1-2 illustrates schematically the analysis approach taken. When selecting this approach, you must have good experience with the physical problem to be solved in order to be able to assign masses to only a few degrees of freedom, and yet obtain an accurate solution to the physical problem considered.
- ▶ Substructuring in dynamic analysis using ADINA can also be employed without "lumping masses" to certain degrees of freedom. In this case all degrees of freedom are treated as mass degrees of freedom (corresponding to a lumped mass matrix), and exactly the same solution is obtained as when no substructuring is employed. The only difference in a solution without substructuring is the cost of the numerical solution.

ref. K.J. Bathe and S. Gracowski, "On Nonlinear Dynamic Analysis Using Substructuring and Mode Superposition," *J. Computers and Structures*, Vol. 13, pp. 699-707, 1981.

This solution scheme can be very effective when large systems need to be solved. Note again that in this case no error is introduced in the solution using the substructuring.

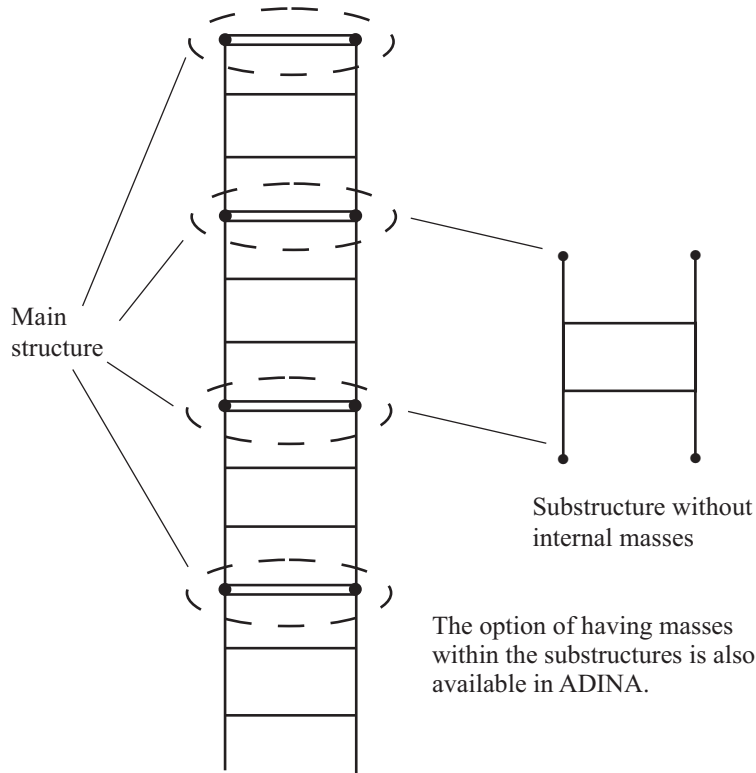


Figure 11.1-2: Substructuring with no masses internal to substructures

11.1.2 Substructuring in nonlinear analysis

- The remarks made regarding substructuring in linear analysis are also applicable when considering the nonlinear analysis of a structural system.
- An important additional consideration (beyond those for linear analysis) is that it can be particularly effective to perform substructuring in a nonlinear analysis when there are only local nonlinearities. In such case, the linear degrees of freedom can be allocated to the linear substructures and the nonlinear degrees of freedom to the master structure. This way the equilibrium iteration is performed in an effective manner.

Some applications are presented in the following reference.

- ref. K.J. Bathe and S. Gracewski, "On Nonlinear Dynamic Analysis Using Substructuring and Mode Superposition," *J. Computers and Structures*, Vol. 13, pp. 699-707, 1981.

11.1.3 Substructuring restrictions

The following restrictions apply to substructures:

- Cyclic symmetric analysis cannot be performed if substructures are present.
- FSI/TMC analysis cannot be performed if substructures are present.
- Mapping analysis options cannot be performed if substructures are present.
- Nonlinear element groups cannot be employed by the substructure, i.e. the material model must be linear, the displacements must be small and the element birth/death option must not be activated. The main structure can, however, consist of both linear and nonlinear element groups.
- Contact surfaces cannot be formed by nodes belonging to a substructure.
- Any midsurface director vector sets that are manually defined must be defined separately for each substructure. Skew systems cannot, however, be defined separately for each substructure since the skew systems defined for the main structure can also be referenced by any of the substructures.
- When substructures are used, rigid links or constraint equations cannot be employed by any substructure or between substructures and the main structure. Hence, constraint equations can only be employed within the main structure.

- Potential-based fluid element groups cannot be employed by the substructure.
- User-supplied element groups cannot be employed by the substructure.
- Substructure loads can only consist of concentrated loads, element pressure loads, prescribed displacements and mass proportional loading. Thus, centrifugal, electromagnetic, temperature loading and pipe internal pressure loading cannot be employed by a substructure.
- Frequency analysis (and consequently mode superposition analysis, response spectrum analysis, harmonic or random vibration analysis) cannot be carried out when substructures are present.
- Buckling eigenvalues and mode shapes cannot be calculated when substructures are present.
- Substructures cannot be used in explicit dynamic analysis.
- Coupling nodes of a substructure must have the same relative locations and the nodal degrees of freedom must have the same directions as for the corresponding main structure nodes. If skew systems or director vectors are used for the coupling nodes, then care must be exercised so that the corresponding skew systems or director vectors for the main structure nodes result in the same directions for the nodal degrees of freedom.
- Substructures must be used with a lumped mass assumption.
- The iterative solver, multigrid solver and 3D-iterative solver cannot be used when substructures are present.
- Load penetration and pressure update (i.e., load stiffening effect of shells) cannot be used when substructures are present.

11.1.4 Additional substructuring modeling techniques

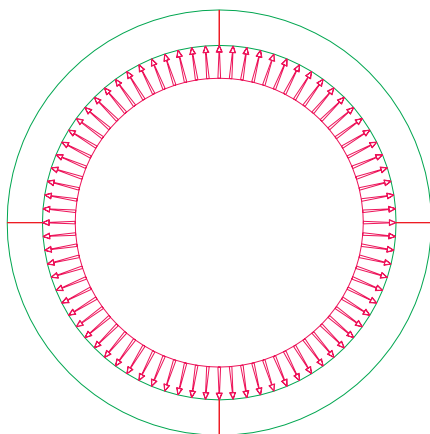
- The effectiveness of substructure modeling is significantly improved when the ADINA substructure/reuse feature is used on

ADINA-M geometry entities. By so doing, the connections between master structure and substructure parts can be easily defined from all AUI/ADINA-M geometry entities, along with meshing options of the commands COPY-TRIANGULATION, FACE-LINK and EG-SUBSTRUCTURE, etc. It is particularly effective when only some local nonlinearities (e.g., contact regions) are considered in a large system analysis: many linear deformation regions can thereby be condensed out and represented just by a few degrees of freedom using the substructure/reuse feature.

In the following examples, we demonstrate the effectiveness of substructure modeling using the substructure/reuse feature.

- Fig. 11.1-3 shows a structure with rotational symmetry (about the X-axis). We can use 4 (or more) substructures to model the entire structure. Substructure/reuse parts are meshed with shell elements and the main structure is represented by isobeam elements. The line loads are applied to inner isobeam elements and fixities are then applied to the outer isobeam nodes. The analysis result will be identical if the model is represented by one master structure.

master structure with isobeam elements



substructure/reuse with shell elements

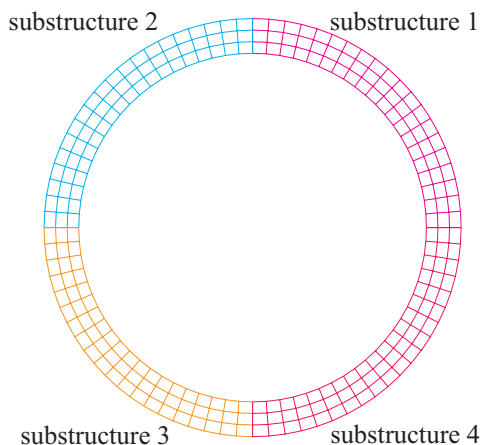


Figure 11.1-3. Use of substructure/reuse feature: model with rotational symmetry (1)

- Fig. 11.1-4 shows another structure with rotational symmetry. Again, we use a 4-substructure representation to model the whole structure. In this example, we assign a plastic material to the isobeam elements which represent the master structure. Shell elements in the substructure/reuse parts are assigned by the linear elastic material. This analysis thus becomes nonlinear. We then verify this result with a model represented by one master structure only. This example shows how local nonlinearities can be handled by substructure modeling in ADINA, a method that can be very effective when dealing with a large finite element analysis model with some local nonlinearities.

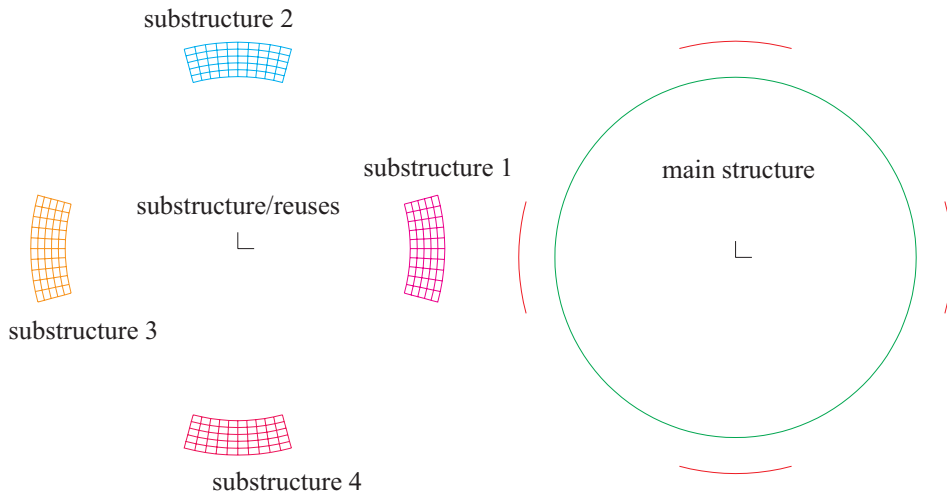


Figure 11.1-4. Use of substructure/reuse feature: model with rotational symmetry (2)

- Fig. 11.1-5 shows a substructure/reuse model used to perform a dynamic analysis on a building. Substructure/reuse parts are meshed using linear shell elements and the reinforced columns/beams are modeled by nonlinear isobeam elements. The impact load is applied at the top left corner node. The analysis is also performed in another model using a master structure only (left half of the same figure). The analysis results from these two models are identical.

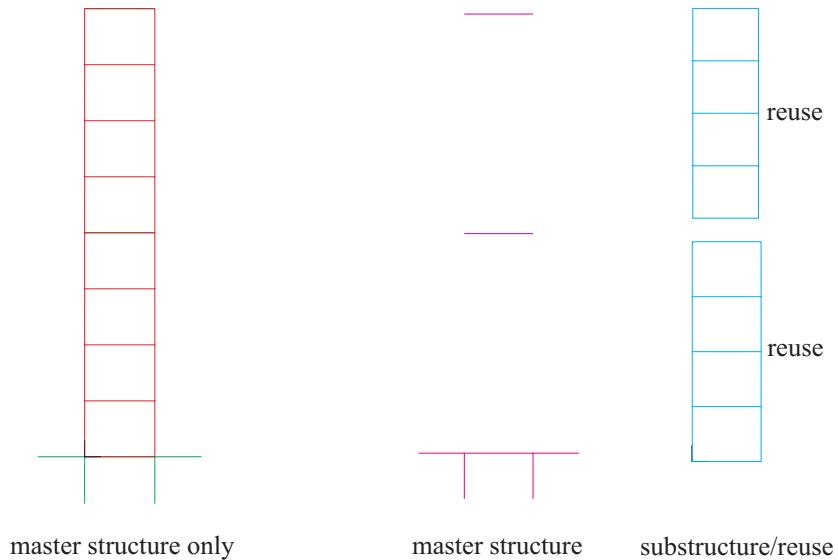


Figure 11.1-5. Use of substructure/reuse feature: model for dynamic analysis of a building

- Fig. 11.1-6 shows a simple model with wave propagation in a rod. The rod is represented by five 3-D brick elements in one substructure. The top and bottom surfaces are represented by shell elements in the master structure. We define the master structure element group in order to define the impact load and the initial condition at the top and to define the boundary condition at the bottom.
- Fig. 11.1-7 shows a 2-D planar contact analysis model. Substructures can model the regions which have no significant deformation — where coarse meshes can be used. The regions near contact surfaces should be represented by the master structure, and fine meshes near contact surfaces need to be used to perform the nonlinear calculations (e.g., contact search) more accurately. The connections between the substructure and the master structure are defined by slave geometry entities and master geometry entities via REUSE command input. The labels ‘master I’ and ‘master II’ denote the different material properties can be used.

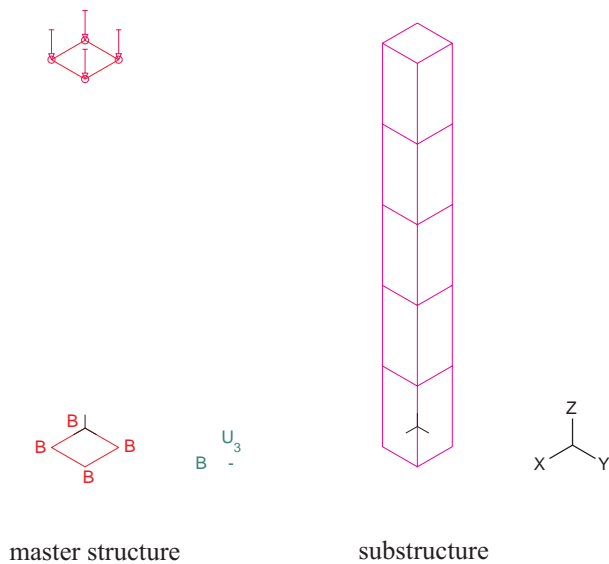


Figure 11.1-6. Use of substructure/reuse feature: wave propagation in a rod

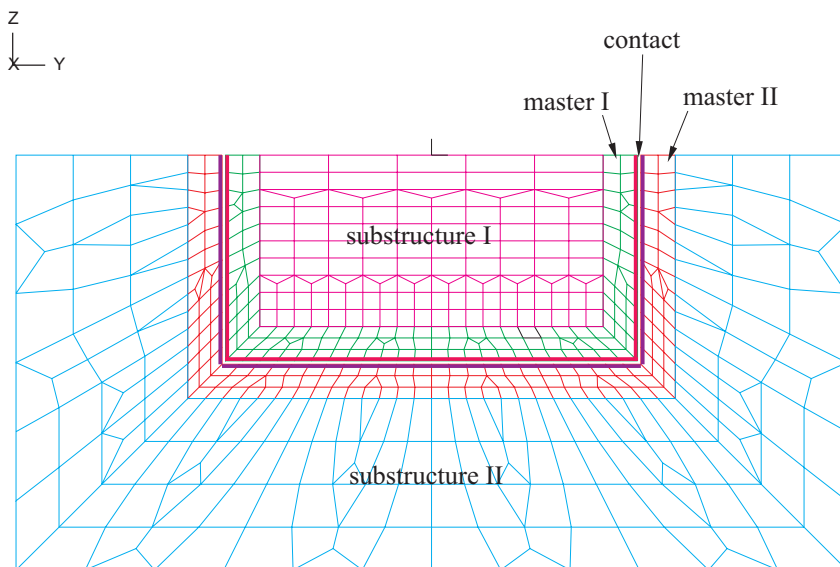


Figure 11.1-7. Use of substructure/reuse feature: 2-D contact example

- Fig. 11.1-8 shows a 3-D contact analysis model, built from ADINA-M solid modeling. This example illustrates how the main structure and the substructure are built for this type of nonlinear analysis. In the actual analysis, finer meshes should be considered near the contact surfaces. The connections between the substructure and the master structure are defined by the REUSE command as in the 2-D example, after which the mesh compatibility between the substructure face and the master face should be enforced by the COPY-TRIANGULATION or FACE-LINK command.

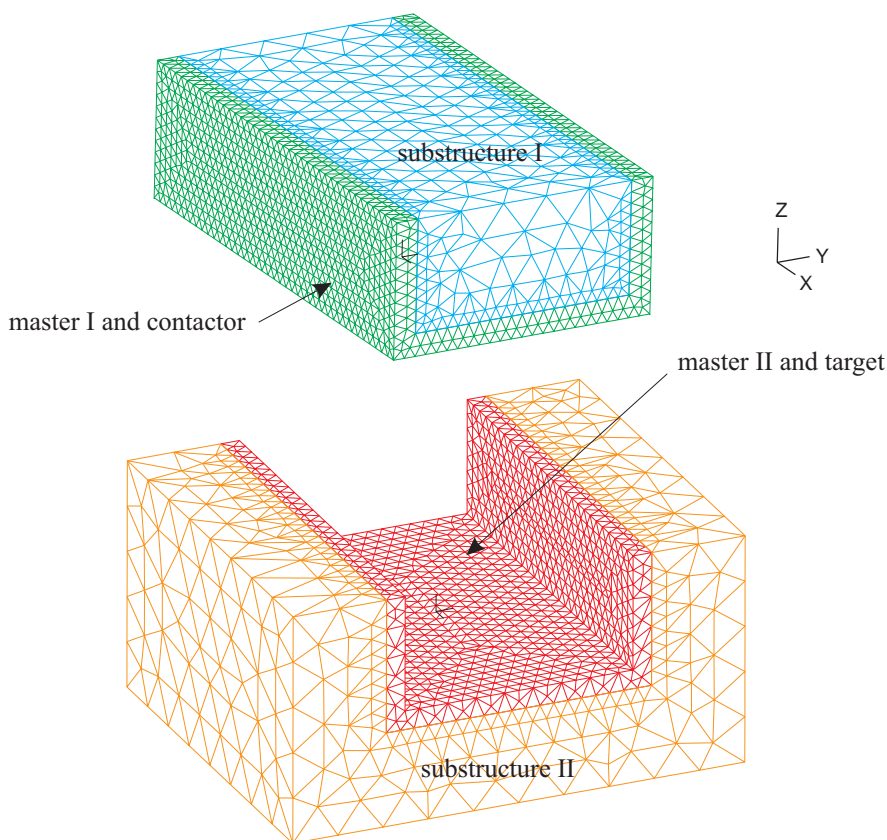


Figure 11.1-8. Use of substructure/reuse feature: 3-D contact example

- Fig. 11.1-9 shows a long columnar block built from one AUI geometry volume and 10 reuses of the meshed volume. Due to the limitations of the current analysis, it is required that dummy shell elements (Youngs modulus=0.001) are defined on the top/bottom surfaces in order to define boundary conditions and loadings. The input steps of Fig. 11.1-9 are as follows:

1. Define Volume 1 (top-surface, label 1 and bottom-surface, label 6)
2. Define Surfaces 101 to 111 and Transformation translations 1 to 11 with direction vectors along column direction
3. SUBSTRUCTURE 1 / EGROUP THREEDSOLID 1 /
GVOLUME 1
4. REUSE 1 / '1' '102' 'surface',,, 'matching' 'transform_1'
'6' '101' 'surface',,, 'matching' 'transform_1'

REUSE 2 / '1' '103' 'surface',,, 'matching' 'transform_2'
'6' '102' 'surface',,, 'matching' 'transform_2'

REUSE 3 / '1' '104' 'surface',,, 'matching' 'transform_3'
'6' '103' 'surface',,, 'matching' 'transform_3'

REUSE ...

REUSE 10 / '1' '111' 'surface',,, 'matching' 'transform_10'
'6' '110' 'surface',,, 'matching' 'transform_10'
5. SUBSTRUCTURE 0 / EGROUP SHELL 1 /
GSURFACE 111 / GSURFACE 101
6. FIXITIES Surface / 111 'all'
7. LOAD force fx=1.0 / APPLY-LOAD / 1 force 1 'surface'
101

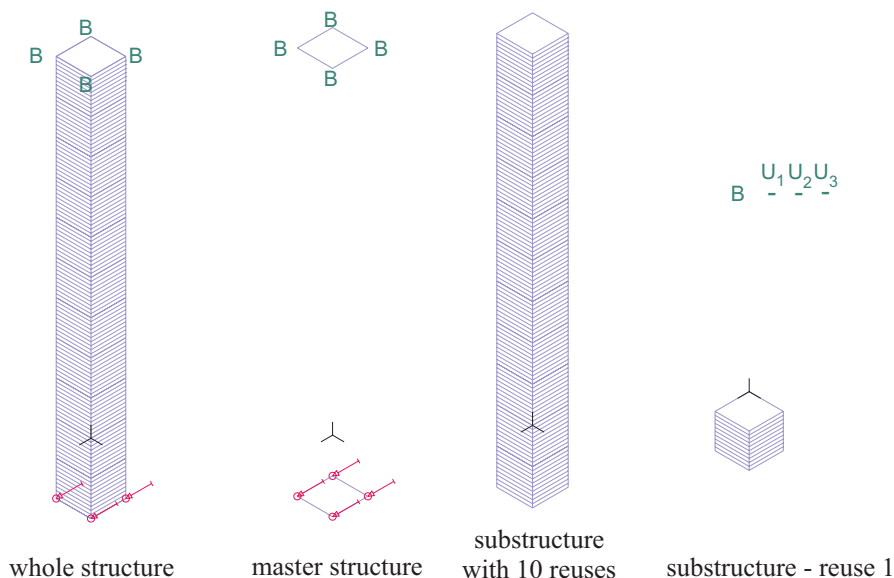


Figure 11.1-9. Use of substructure/reuse feature: column substructure with 10 reuses

- Some modeling considerations and limitations:
 1. Shell node degrees of freedom are determined as described in Section 2.7.3 for the master structure and the substructure shell element nodes with the SHELLMODE = AUTOMATIC option in the MASTER command, see Fig. 11.1-10.
 2. Nodes at the connection locations are always assigned global six degrees of freedom for the shell nodes, see Fig. 11.1-10.
 3. To input boundary conditions and loadings at the master structure nodes requires the element definitions at these master nodes, see Fig. 11.1-9.
 4. The same material property is assigned in the various REUSE definitions of the same substructure.
 5. The option TRANSFORMATION in the REUSE command can only be a pure translation type, as shown in Fig. 11.1-5, Fig. 11.1-9, and Fig. 11.1-10.

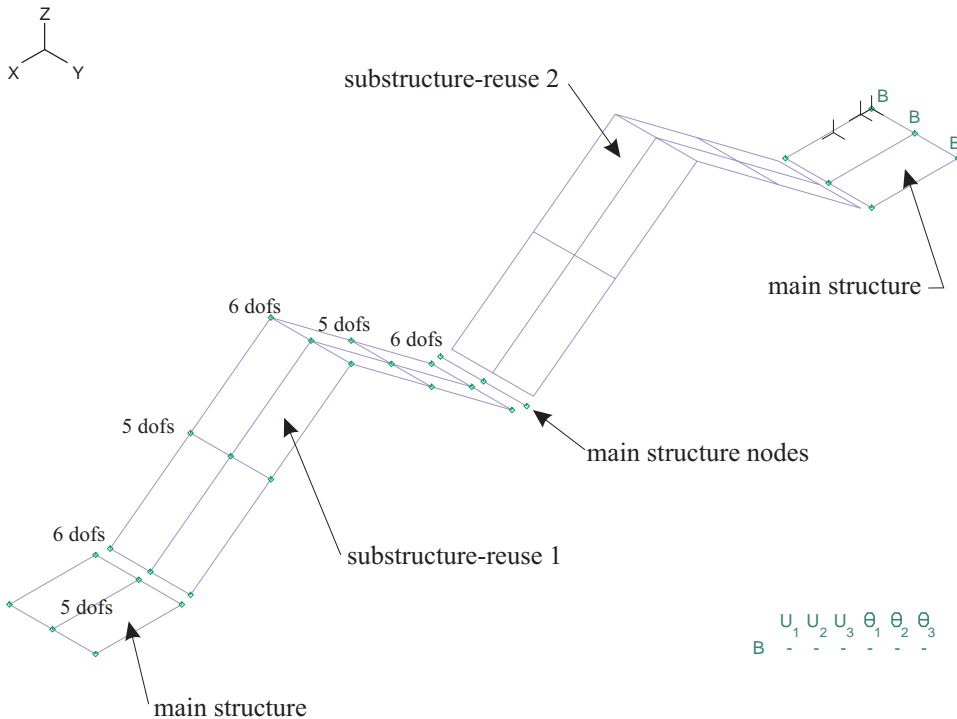


Figure 11.1-10. Use of substructure/reuse feature: assignment of degrees of freedom to nodes at connection locations

11.2 Cyclic symmetry analysis

- Cyclic symmetry analysis is useful for components when the geometry and boundary conditions are rotationally symmetric. One fundamental cyclic part is modeled and the complete structure is obtained by rotating the fundamental part N times about an axis of cyclic symmetry, where N is the number of cyclic parts. Figure 11.2-1 shows a schematic example.
- Note that there is no approximation in the analysis when the cyclic symmetry option is used: the same result is obtained as if the complete structure is modeled.

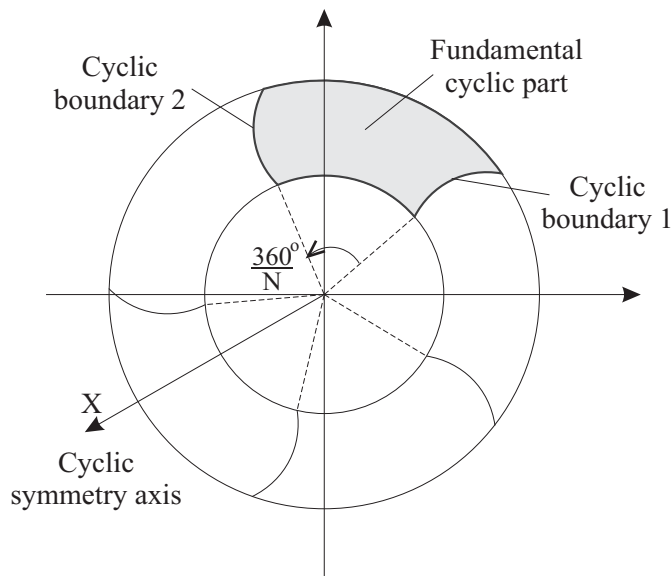


Figure 11.2-1: Schematic of a cyclically symmetric component

- For a model with N cyclic parts, instead of solving for the complete model at once, the solution is divided into M harmonics, where

$$M = \begin{cases} \frac{N+2}{2} & \text{if } N \text{ is even} \\ \frac{N+1}{2} & \text{if } N \text{ is odd} \end{cases}$$

Each harmonic is solved separately, and the results are superimposed to obtain the solution for the composite model.

- For more details on the theory involved in cyclic symmetry, see

ref. W. Zhong and C. Qiu, "Analysis of Symmetric or Partially Symmetric Structures," *Computer Methods in Applied Mechanics and Engineering*, Vol. 38, pp. 1-18, 1983.

- Although the geometry, material properties and displacement

boundary conditions must be cyclically symmetric, the external applied loads can be arbitrary, including pressure loads, concentrated loads, centrifugal and mass proportional loading, etc. Prescribed displacement degrees of freedom have to be the same on all cyclic parts. However, the amplitude of the prescribed displacement can vary from one part to the next (see Figure 11.2-2).

- Note that “solid” cyclic structures with nodes along the cyclic symmetry axis can also be modeled.
- If the loads are also cyclically symmetric, the model also possesses *periodic symmetry*. This assumption further simplifies the analysis (see Figure 11.2-2). Note that this assumption corresponds to having all the loads on the first harmonic (harmonic 0). In this case, only solution of the first harmonic is required.
- For general cyclic symmetry (not periodic symmetry), linear static, linear dynamic, and frequency analysis can be performed. In linear dynamic analysis, mode superposition or implicit time integration can be used.

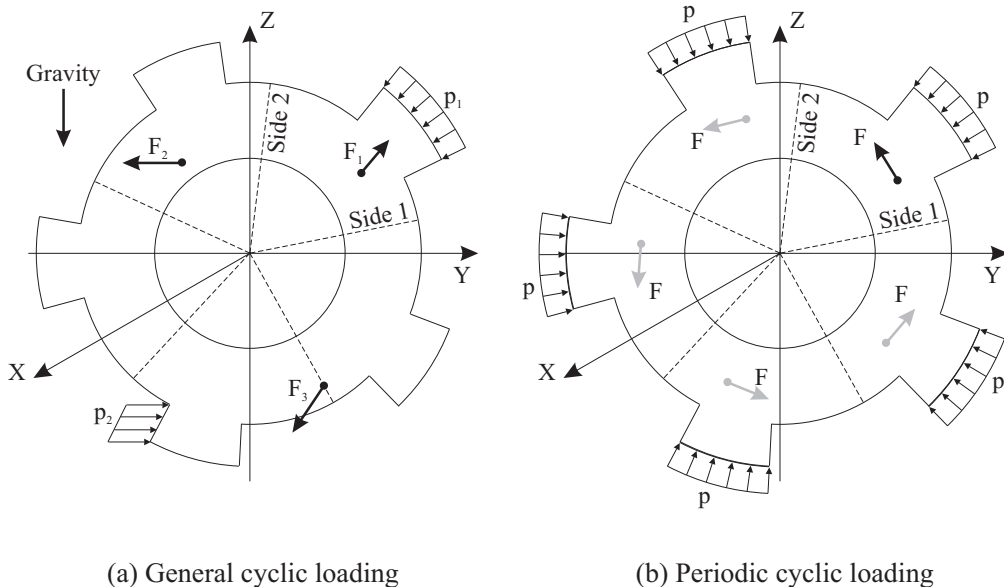


Figure 11.2-2: Applied loads in cyclic symmetry analysis

- For periodic symmetry, a nonlinear analysis can be performed as well (including large deformation, large strains, contact, etc.). A periodic frequency analysis only obtains the periodic natural frequencies. Thermomechanical coupling (TMC) problems can also be set up with period symmetry.
- In addition to the sparse solver, the 3D-iterative solver can also be used with periodic symmetry. The 3D-iterative solver, however, cannot be used with cyclic symmetry.
- A nonlinear periodic static or dynamic analysis can be followed by a full cyclic symmetry frequency analysis to obtain all the natural frequencies and mode shapes of a the deformed structure.
- In cyclic frequency analysis, the user can specify either the total number of frequencies (default) or the number of frequencies per harmonic. If the total number is specified, ADINA will internally determine how many frequencies to solve per harmonic. A conservative (large) value will be used to ensure that no frequency is skipped.
- Also in cyclic frequency analysis, the complete range of harmonics can be used (default), or alternatively, the lowest and highest harmonic values can be set by the user.
- Another feature in cyclic frequency analysis is to solve for all frequencies, or just solve for one a every complex frequency pairs. This is useful because for all except the harmonic 0 (and the last harmonic is N is even), cyclic frequencies come in pairs.
- The cyclic symmetric analysis option has three advantages:
 - ▶ Only a small part of the complete structure needs to be modeled.
 - ▶ For general cyclic symmetry, the storage required is reduced up to a factor of $N/2$, where N is the number of cyclic parts. For periodic symmetry, the storage required is reduced up to a factor of N .
 - ▶ The solution time can be significantly less than when the complete structure is modeled. As an estimate, the cyclic symmetric analysis will reduce the solution time by a factor of

up to $N/2$.

- Only the finite element model of the fundamental cyclic part should be created with the AUI. The displacement boundary conditions are only defined on the fundamental cyclic part. Note that the displacements on the lines and/or surfaces making up the cyclic boundaries are unknown (unless they are fixed or prescribed).
- The loads on each of the N cyclic parts must be defined. Loads on a cyclic boundary can be input as loads onto either of the adjacent cyclic parts.
- Results are saved for all cyclic parts, except in periodic symmetry when they are only saved for the fundamental cyclic part. When loading results, the user can select to load only a subset of cyclic parts. This feature reduces the memory required for post-processing cyclic symmetry results.
- Cyclic symmetry constraints are automatically generated on the cyclic boundary. Therefore, the degrees of freedom on the cyclic boundary cannot be constrained by rigid links or constraint equations.
- The nodes of cyclic boundary 1 must coincide with those of cyclic boundary 2 when rotated counterclockwise $360/N$ degrees about the axis of cyclic symmetry. In 2-D analysis, this is achieved by using the same mesh divisions along both cyclic boundaries. The same applies to 3-D analysis with mapped meshes. In 3-D analysis with free-form meshing and 3-D solid elements, using the same mesh divisions on both cyclic boundaries will not guarantee that a matching mesh is created throughout the boundary. To ensure identical meshes on both cyclic boundaries, create a dummy shell mesh on one cyclic boundary, then copy the resulting mesh pattern to the other cyclic boundary via the “Copy Mesh Triangulation” command, then create the 3-D mesh, and delete the dummy shell element mesh.
- The nodes of cyclic boundary 2 are automatically constrained to those of cyclic boundary 1 via the constraints based on the theory of cyclic symmetry. For this reason, nodes on cyclic boundary 2

cannot be otherwise constrained (by other constraints or rigid links).

- The axis of cyclic symmetry can have any spatial orientation.
- Note that if structural elements (truss, beam, iso-beam, plate, shell or pipe elements) are defined exactly on the cyclic boundaries, then the complete structural elements must be input in the definition of the fundamental cyclic part.
- Since a cyclic boundary connects two adjacent cyclic parts (for one cyclic part it will be cyclic boundary 1 and for the other, cyclic boundary 2), the loads on a cyclic boundary can be input as loads onto either of the adjacent cyclic parts. However, prescribed displacements on a cyclic boundary must be applied on the cyclic part when it is cyclic boundary 1.
- In dynamic analysis, the initial conditions for the N cyclic parts must be defined.
- In frequency analysis, the eigenvectors output by the cyclic symmetry procedure are not M-normalized.
- The following analysis options are not possible with cyclic symmetry:
 - ▶ Potential-based fluid elements
 - ▶ Axisymmetric and generalized plane stress 2-D elements
 - ▶ Connector elements
 - ▶ Piping analysis with ovalization/warping degrees of freedom
 - ▶ Substructure analysis
 - ▶ Response spectrum analysis
 - ▶ Harmonic or random vibration analysis
 - ▶ Fracture mechanics analysis
 - ▶ FSI analysis
 - ▶ Explicit time integration
 - ▶ TMC analysis (unless periodic symmetry)

11.3 Reactions calculation

- Computations of the reaction forces and moments can be requested for all degrees of freedom where fixity boundary conditions are applied and where displacements are prescribed. No reactions are calculated for the degrees of freedom corresponding to deleted master boundary conditions.

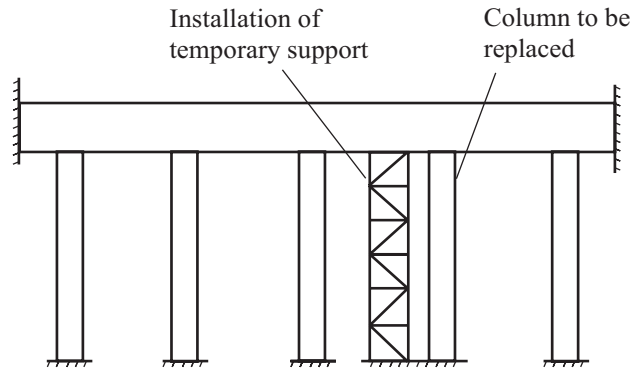
Reaction calculations are on by default.

- The loads applied to fixed or prescribed degrees of freedom do not contribute to the displacement and stress solutions. However, these loads are accounted for in the reaction calculations. This is important to ensure that the sum of the reaction forces balance the sum of the applied loads in static analysis.
- All applied loads are included in the reaction calculations, except for the FSI forces. The FSI forces are output separately as the variable FSI_FORCE-(XYZ). This allows the resultant FSI force acting on the structure to be determined.
- Reaction forces and moments at a node are computed using the consistent force vectors (calculated from the element internal stresses) of elements attached to the node. Hence, a check on the balance of the support reactions and the applied loads often provides a good measure on the accuracy of the solution (in terms of satisfying equilibrium in a nonlinear analysis).
- Reaction calculations in direct time integration dynamic analysis and mode superposition analysis with consistent mass matrix take into account the mass coupling to the deleted degrees of freedom. The reactions exactly equilibrate the applied forces in all cases.
- If Rayleigh damping is used, then the damping contribution to deleted degrees of freedom is taken into account for implicit time integration.

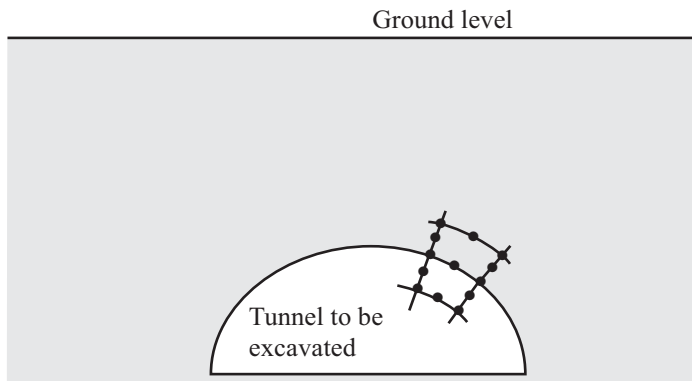
11.4 Element birth and death feature

- The element birth and death feature is useful for modeling processes during which material is added to and/or removed from the structure. Such processes, for example, are encountered in the construction of a structure (structural members are added in succession), the repair of a structure (structural components are removed and new ones are added) or during the excavation of geological materials (a tunnel is excavated). If the element birth and death option is used, the corresponding element groups become automatically nonlinear. Fig. 11.4-1 illustrates two analyses that require the element birth and death options.
- The main features of the element birth and death option are as follows:
 - ▶ If the element birth option is used, the element is added to the total system of finite elements at the time of birth and all times thereafter.
 - ▶ If the element death option is used, the element is taken out of the total system of finite elements at times equal to and larger than the time of death
 - ▶ If both element birth and death options are used, the element is added to the total system of finite elements at the time of birth and remains active until the time of death. The time of death must be greater than the time of birth. The element is taken out of the total system of finite elements at all times equal to and larger than the time of death.
- Once an element is born, the element mass matrix, stiffness matrix and force vector are added to the mass matrix, stiffness matrix and force vector of the total element assemblage (until the death time, if any). Similarly, once an element dies, the element mass matrix, stiffness matrix and force vector are removed from the total assembled mass matrix, stiffness matrix and force vectors for all solution times equal to or larger than the time of death of the element.

- The element birth/death option applies to any mass effect i.e., gravity loading, centrifugal loading and inertia forces. The mass matrix, therefore, does not remain constant throughout the solution.
- Note that the damping matrix is not modified when elements die or when they are born. Therefore, element groups with such elements should have no Rayleigh damping. For example, if you want to simulate a pipe break by setting the death time for certain elements in an implicit dynamic analysis, then the element group containing these elements should have damping parameters set to zero. Otherwise, you might not see a pipe separation.
- When the element birth/death option is used, the tangent stiffness matrix may at some solution times contain zero rows and corresponding columns. The equation solver disregards any zero diagonal element in the tangent stiffness matrix if no elements are attached to the associated degrees of freedom.
- ADINA enables the user to set an element death decay time parameter which causes the gradual reduction of the element force vector and stiffness matrix to zero over a finite time rather than instantly. The reduction starts at the death time and progresses linearly with time until the decay time has passed. The element therefore totally vanishes at a time equal to the sum of the death time and the death decay time. This option is useful for mitigating the discontinuity that the structure may experience due to the death of some of its elements.
- The birth/death feature is also available for contact pairs in contact analysis, and for mesh glued surfaces.
- To provide the appropriate concentrated and element loading that takes into account the element birth/death option, you have to specify time functions on the loading that correspond to the proper elements (however, see also Section 11.5).



(a) Repair of a bridge



(b) Excavation of a tunnel

Figure 11.4-1: Analyses that require the element birth and death options

- The time at which an element becomes active or inactive is specified by the parameters TBIRTH and TDEATH respectively (in the EDATA and EGROU commands).

Birth option active

In the discussion of element birth, it is useful to refer to the “preborn” time. The program determines the preborn time to be the solution time that just precedes the user-input time of element birth.

We will discuss the selection of the preborn time in more detail below.

Now we describe how the element displacements at the preborn time are used in the element strain calculations. Recall that the current element coordinates, original element coordinates and displacements are related by

$${}^{t+\Delta t}\mathbf{x} = {}^0\mathbf{x} + {}^{t+\Delta t}\mathbf{u}$$

and that the element strains are calculated using the displacements and original coordinates, using, for example in geometrically linear analysis,

$${}^{t+\Delta t}\mathbf{e} = \frac{\partial {}^{t+\Delta t}\mathbf{u}}{\partial {}^0\mathbf{x}}$$

with similar calculations in geometrically nonlinear analysis.

For an element that is born, these relationships are modified as follows:

$${}^{t+\Delta t}\mathbf{x} = \left({}^0\mathbf{x} + {}^{pb}\mathbf{u} \right) + \left({}^{t+\Delta t}\mathbf{u} - {}^{pb}\mathbf{u} \right)$$

$${}^{t+\Delta t}\mathbf{e} = \frac{\partial \left({}^{t+\Delta t}\mathbf{u} - {}^{pb}\mathbf{u} \right)}{\partial \left({}^0\mathbf{x} + {}^{pb}\mathbf{u} \right)}$$

in which ${}^{pb}\mathbf{u}$ are the displacements at the preborn time. The quantity ${}^0\mathbf{x} + {}^{pb}\mathbf{u}$ are the coordinates at the preborn time and the quantity ${}^{t+\Delta t}\mathbf{u} - {}^{pb}\mathbf{u}$ are the displacements relative to the coordinates at the preborn time.

From the above, we observe that if the current displacements are the same as the preborn displacements, the element strains are zero. And, if there are no thermal or initial strains in the element, the element stresses are also zero.

Now we discuss in detail how the program determines the preborn time.

Figure 11.4-2(a) shows the activity of an element for which the birth option is active. Note that if TBIRTH is input for the range shown (where $TBIRTH > t + \varepsilon$ and $TBIRTH < t + \Delta t + \varepsilon$), then the preborn time for the element is t and the element is first active at time $t + \Delta t$. Here ε is a program-calculated tolerance, typically $\Delta t / 1000$. The results obtained are independent of the exact position of TBIRTH within the range of solution times $t + \varepsilon$ to $t + \Delta t + \varepsilon$.

Two special cases are noteworthy:

TBIRTH just slightly greater than $t + \varepsilon$, for example $t + \Delta t / 100$. In this case, the preborn time is t , which is close to TBIRTH. Because TBIRTH is close to t , the preborn time is essentially selected by the choice of TBIRTH, and the element can be thought of as being born “strain-free”. The element is first active at time $t + \Delta t$. If the element displacements remain unchanged between times t and $t + \Delta t$, the element remains strain-free. See also the example given below.

$TBIRTH = t + \Delta t$. In this case, the preborn time is t (and not $t + \Delta t$). The element is first active at time $t + \Delta t$, however the element need not be strain-free at time $t + \Delta t$.

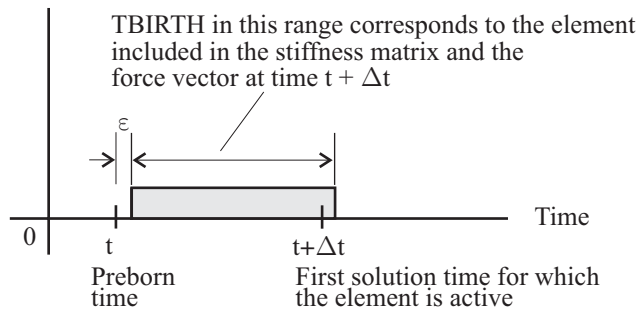
Regarding the mass matrix, when the element is born, the mass matrix for the element is computed using the original coordinates of the element ${}^0\mathbf{x}$, and not the coordinates at the preborn time ${}^0\mathbf{x} + {}^{pb}\mathbf{u}$.

Death option active

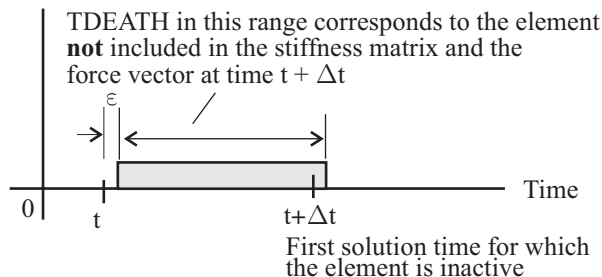
Fig.11.4-2(b) shows the activity of an element for which the death option is active. Note that if TDEATH is input for the range shown (where $TDEATH > t + \varepsilon$ and $TDEATH < t + \Delta t + \varepsilon$), the element is first inactive at time $t + \Delta t$.

Birth then death option active

This is a direct combination of the birth and death options. Initially some elements in an element group are inactive. At a particular solution time determined by the time of birth $TBIRTH$, the elements become active and remain so until a subsequent solution time determined by the time of death $TDEATH$, where $TDEATH > TBIRTH$.



(a) Birth option active



(b) Death option active

Figure 11.4-2: Use of element birth and death option

Example of the element birth option

Consider the materially linear truss element model shown in Fig. 11.4-3(a) in which the time of birth for element 2 is slightly larger than t , e.g. $TBIRTH = t + \Delta t / 100$. Element 2 is unborn at time t , as shown in Fig. 11.4-3(b).

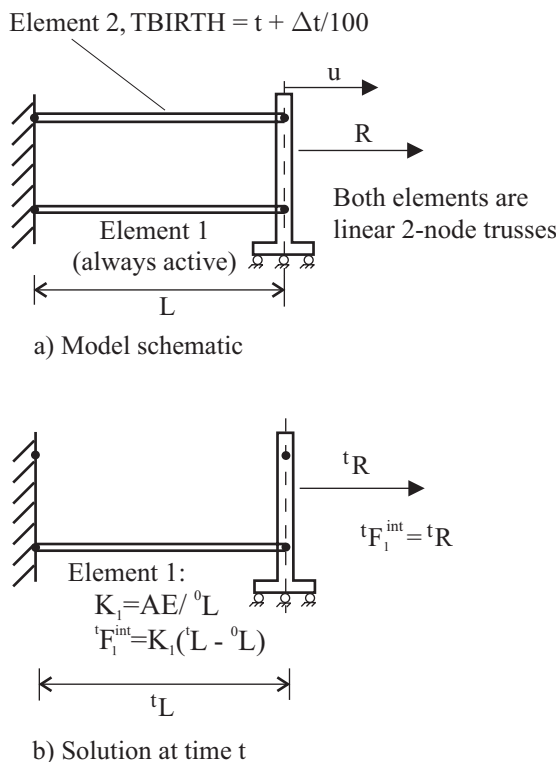


Figure 11.4-3: Example on the use of the element birth option

At the beginning of the solution for time $t + \Delta t$, the program determines that element 2 is active for time $t + \Delta t$, and that the preborn time t_{pb} is equal to the solution time t . The program stores ${}^{pb}u = {}^tu$. In the subsequent equilibrium iterations, the element relative displacement is ${}^{t+\Delta t}u - {}^{pb}u = {}^{t+\Delta t}u - {}^tu$, the

element length is ${}^0L + {}^{pb}u = {}^0L + {}^t u = {}^t L$ and the element strain is ${}^{t+\Delta t}e = \frac{{}^{t+\Delta t}u - {}^{pb}u}{{}^0L + {}^{pb}u}$. This procedure is physically identical to

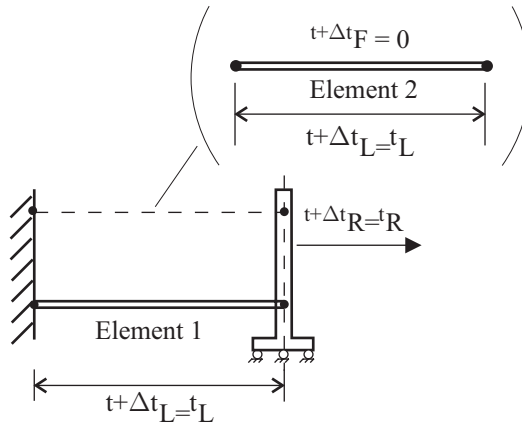
adding an element of length ${}^t L$ to the assemblage at time $t + \Delta t$.

Note that the stiffness of element 2 is based on the length ${}^t L$.

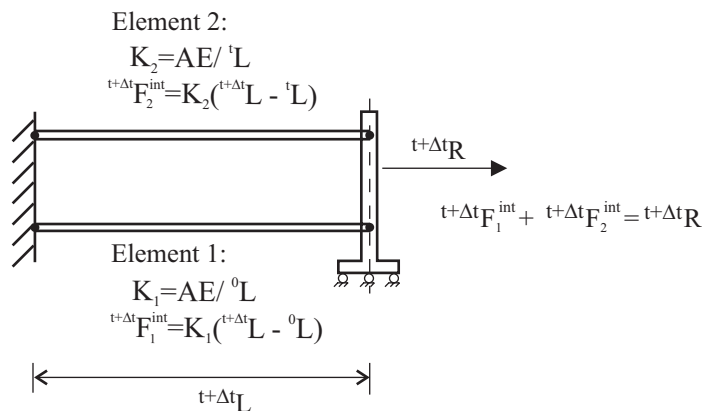
If the external force ${}^{t+\Delta t}R = {}^t R$, then element 2 is stress-free, as the system was in equilibrium at time t without any forces from element 2 (Fig 11.4-3(c)).

If the external force ${}^{t+\Delta t}R \neq {}^t R$, then element 2 is not stress-free, and the force in element 2 is determined based on its deformation with respect to its preborn state (the solution at time t), as shown above (Fig 11.4-3(d)).

Hence, the total increment in displacement from time t to time $t + \Delta t$ determines the force in element 2. Identically, the same solution would be obtained using any value for TBIRTH which satisfies the relation $\text{TBIRTH} > t + \varepsilon$ and $\text{TBIRTH} < t + \Delta t + \varepsilon$.



c) Solution at time $t+\Delta t$, ${}^{t+\Delta t}\mathbf{R} = {}^t\mathbf{R}$



d) Solution at time $t+\Delta t$, ${}^{t+\Delta t}\mathbf{R} \neq {}^t\mathbf{R}$

Figure 11.4-3 (continued)

11.5 Element "death upon rupture"

- For the materials and elements listed in Table 11.5-1, element death is automatically activated when rupture is detected at any integration point of the element. The element is then considered as "dead" for the remainder of the analysis, and, in essence, removed from the model (mass and stiffness contributions).
- As elements become dead, contactor segments connected to

these elements are also removed from the model.

- Load penetration can be activated to ensure that pressure loads are properly redistributed inside voids created by "dead" elements (see Section 5.3.1).

Table 11.5-1: Elements and material models that include "death upon rupture"

	Truss*	2-D solid	3-D solid	Beam	Iso- beam	Shell	Pipe
Plastic-bilinear, plastic-multilinear	✓	✓	✓	-	✓	✓	✓
Mroz-bilinear	-	✓	✓	-	-	-	-
Thermo-plastic, creep, plastic-creep, multilinear-plastic-creep, creep-variable, plastic-creep-variable, multilinear-plastic-creep-variable	✓	✓	✓	-	✓	✓	✓
Plastic-orthotropic	-	✓	✓	-	-	✓	-
Plastic-cyclic	✓	✓	✓	✓	-	✓	-
Moment-curvature	-	-	-	✓	-	-	-
User-coded	-	✓	✓	-	-	-	-

*The rupture option is not applicable to a truss element with a gap.

11.6 Initial conditions

11.6.1 Initial displacements, velocities and accelerations

- You can specify the initial displacements, velocities and accelerations of the nodes.
- Any initial displacements, velocities and accelerations specified in a restart run are ignored, except when restarting from a static to a dynamic analysis; in this case, initial velocities are taken into account.

- Initial rotations should only be applied in small displacement analysis.
- The initial rotations at a node are interpreted differently depending upon the element attached to the node.
 - ▶ Element without rotational dofs: rotations are ignored
 - ▶ Linear or MNO element with rotational dofs: initial rotations are used in the calculation of the initial force vector.
 - ▶ Large displacement element with rotational dofs: initial rotations are ignored.
 - ▶ Small displacement rigid link: the initial position of the slave node is determined by the initial rotation of the master node.
 - ▶ Large displacement rigid link: the initial position of the slave node is obtained using the initial rotation of the master node, as if the rigid link is a small displacement rigid link. This will cause the length of the rigid link to be incorrect if the initial rotation is large.

11.6.2 Initial temperatures and temperature gradients

- The formulas used to determine the initial nodal temperatures are as follows:

Initial temperature explicitly specified: ${}^0\theta_i = {}^0\theta_i^E$

Initial temperature not explicitly specified: ${}^0\theta_i = \text{TINIT}$

in which ${}^0\theta_i$ is the initial nodal temperature at node i , TINIT is the overall initial temperature (also referred to as the default initial temperature) and ${}^0\theta_i^E$ is the explicitly specified initial temperature at the node.

Initial temperatures can also be read from a temperature file. In this case, any directly specified initial temperatures are added to those read from the temperature file.

Initial temperature gradients are handled in an analogous manner. These are only used at axisymmetric shell nodes and at shell mid-surface nodes in order to calculate initial temperatures in these elements. Non-zero initial temperature gradients cannot be applied to shell nodes.

Initial temperatures and temperature gradients specified in a restart run are ignored.

- When temperatures are directly prescribed (as in Section 5.6), the directly prescribed temperatures are not used to determine the initial temperatures. Initial temperatures must be specified as described above.
- The thermal strains are always assumed to be zero initially, see Section 3.1.4.

11.6.3 Initial pipe internal pressures

- The initial pipe internal pressures consist of the sum of the explicitly defined pipe internal pressure loads (evaluated at time TSTART) and the pipe internal pressures read from file, if any. If there are no explicitly defined pipe internal pressure loads and no pressure loads read from file, the initial pipe internal pressures are set to zero.

Initial pipe internal pressures specified in a restart run are ignored.

11.6.4 Initial strains

- When the initial strain feature is used, the strains used in the stress-strain law include the initial strains. Symbolically

$$\boldsymbol{\tau} = \boldsymbol{\tau}(\mathbf{e} + {}^0\mathbf{e})$$

where $\boldsymbol{\tau}$ are the stresses, the function $\boldsymbol{\tau}()$ is the function that computes stresses from strains, \mathbf{e} are the strains computed from the displacements and ${}^0\mathbf{e}$ are the initial strains. For example, for a linear elastic material

$$\boldsymbol{\tau} = \mathbf{C}(\mathbf{e} + {}^0\mathbf{e})$$

where \mathbf{C} is the stress-strain matrix.

In the above, we assume that a materially-nonlinear-only formulation is used, we will discuss large displacement formulations later.

- Figure 11.6-1 shows a single 2-D element in which the initial strain option is used, and in which no other applied loads or body forces are present. Tensile initial strains are applied. When the element is unrestrained, $\boldsymbol{\tau} = \mathbf{0}$, hence $\mathbf{0} = \mathbf{e} + {}^0\mathbf{e}$, hence \mathbf{e} is compressive, corresponding to an element that has shrunk. When the element is rigidly restrained from motion, $\mathbf{e} = \mathbf{0}$, hence the element contains tensile stresses.

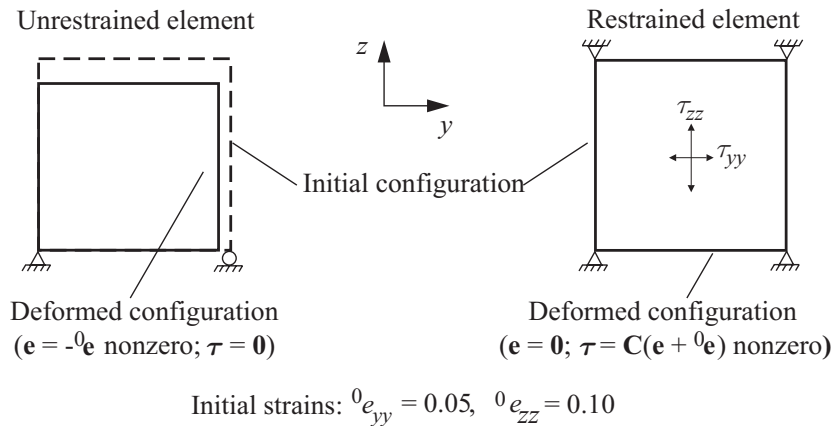


Figure 11.6-1: Initial strains applied to 2-D solid element

This example illustrates the following general principles:

- ▶ Tensile initial strains applied to an unrestrained assemblage of elements cause the assemblage to shrink.
- ▶ Tensile initial strains applied to a restrained assemblage of elements induce tensile stresses in the assemblage.
- The initial strain option is available for all solid and structural elements, but only for nonlinear element groups. If the initial strain option is applied to an element group that would otherwise be linear, the element group becomes nonlinear.
- All material models except for the Ilyushin material model can be used.
- The strains printed or saved to the porthole file include the

initial strains, that is $\mathbf{e}_{output} = \mathbf{e} + {}^0\mathbf{e}$.

Initial strain axes

In general, initial strain components are interpreted in an initial strain axes coordinate system. These axes are denoted (1-2-3), so the initial strain components are ${}^0e_{11}$, ${}^0e_{22}$, ${}^0\gamma_{12}$, etc. Notice that the shear strain components use the engineering convention.

The initial strain axes for the truss, beam, iso-beam and pipe elements have the “1” axes aligned with the axial direction of the element. In most cases, only the axial initial strain ${}^0e_{11}$ is considered, however for the axisymmetric shell element, the hoop strain component ${}^0e_{33}$ is also considered.

The initial strain axes for the 2-D solid, 3-D solid, plate and shell elements are user-defined. For the 2-D solid, plate and shell elements, the 1-2 axes must lie in the plane of the element and the 3 axis lies out of the plane of the element. One way to define initial strain axes is to use the AXES commands of the AUI. In the following example, axes-system 27 is defined, then the initial strain axes for the elements on surface 3 are set to axes-system 27:

```
AXES CONSTANT 27,
      AX=0.5 AY=0.3 AZ=0.2 BX=-0.5 BY=0.2 BZ=0.1
SET-AXES-STRAIN SURFACES
ENTRIES SURFACE AXES
      3      27
DATAEND
```

Input of initial strains

The initial strains can be entered as nodal point initial strains, element initial strains or a combination of nodal point and element initial strains.

Nodal point initial strains

Nodal point initial strains can be specified as follows:

- At the nodes, using the INITIAL STRAINS command.

- On geometry, using the INITIAL-CONDITION command to create an initial condition, then using the SET-INITCONDITION command to assign the initial condition to the geometry.

The nodal point initial strains are specified in terms of initial strain components in the initial strain axes coordinate system. For plate elements, the membrane and flexural initial strains can be specified, and for the shell elements, the membrane and gradient initial strains can be specified.

Within each element, the nodal point initial strain components are interpolated, component by component, to the integration points.

Because the initial strain axes can have different orientations in different elements, the nodal point initial strain components can be interpreted differently in attached elements. For example, suppose that node 45 has ${}^0e_{11} = 0.02$, and suppose that node 45 is attached to elements 27 and 65. Within element 27, the value ${}^0e_{11} = 0.02$ is interpreted according to the initial strain axes of element 27, and within element 45, the value ${}^0e_{11} = 0.02$ is interpreted according to the initial strain axes of element 45.

Element initial strains — truss, beam, iso-beam, shell, pipe elements

For the truss, beam, iso-beam, shell and pipe elements, the element initial strain components are specified as element data (and possibly also assigned to the geometry). For example, to assign membrane element initial strains to shell elements, commands similar to

```
* SHELL ELEMENT
EDATA
ENTRIES EL EPS11 EPS22 EPS12 EPS13 EPS23
        1  0.01  0.02 -0.01 -0.02 -0.03
DATAEND
```

and

```
SURF-ELEMDATA SHELL
ENTRIES SURFACE EPS11 EPS22 EPS12 EPS13 EPS23
```

```

2      0.01  0.02 -0.01 -0.02 -0.03
DATAEND

```

can be used.

Element initial strains — 2-D solid element (except for the 3D plane stress element)

For the 2-D solid element, the element initial strain components are specified using strain field coefficients A, B, C, D, E, F and the formulas

$$\begin{aligned}
 {}^0e_{zz} &= A + Bz \\
 {}^0e_{yy} &= C {}^0e_{zz} + D \\
 {}^0e_{xx} &= E {}^0e_{zz} + F \text{ (axisymmetric and generalized plane strain} \\
 &\text{elements only)}
 \end{aligned}$$

The dependence of the initial strains on the z coordinate is useful for modeling initial strains in geological materials.

The strain field coefficients are defined using the AUI command STRAIN-FIELD.

The subscripts x, y, z refer to the 2D coordinate system after modification by the MASTER 2DPL-AX flag. If MASTER 2DPL-AX=YZ-Z (the default), the 2-D elements all lie in the global y - z plane with the global z axis as the axisymmetric axis. In this case the 2D coordinate system is the same as the global system and the above formulas give the initial strains in the global system. However if, for example, MASTER 2DPL-AX=XY-Y, then the 2-D elements all lie in the global x - y plane with the global y -axis as the axisymmetric axis. In this case, the y axis of the 2D system is the global x axis and the z axis of the 2D system is the global y axis. So then in the above equations z refers to the global y coordinate, ${}^0e_{zz} = A + Bz$ refers to the normal initial strain acting in the global y direction, ${}^0e_{yy} = C {}^0e_{zz} + D$ refers to the normal initial strain acting in the global x direction and ${}^0e_{xx} = E {}^0e_{zz} + F$ refers to the normal initial strain acting in the global z direction.

Element initial strains — 2-D solid element, 3D plane stress option

For the 2-D solid element using the 3D plane stress option, the element initial strain components are specified using strain field coefficients A , B , C , D and the formulas

$$\begin{aligned} {}^0e_{zz} &= A + Bz \\ {}^0e_{yy} &= C {}^0e_{zz} + D \end{aligned}$$

Here variable z is the global z coordinate, however subscripts y , z refer to the element local y and z directions.

Element initial strains — 3-D solid element

For the 3-D solid element, the element initial strain components are specified using strain field coefficients A , B , C , D , E , F and the formulas

$$\begin{aligned} {}^0e_{zz} &= A + Bz \\ {}^0e_{xx} &= C {}^0e_{zz} + D \\ {}^0e_{yy} &= E {}^0e_{zz} + F \end{aligned}$$

See comments for the 2-D solid element concerning the strain field coefficients. For the 3-D solid element, variable z is always the global z coordinate, and subscripts x , y , z always refer to the global coordinate system.

Summary of initial strain input

Table 11.6-1 gives a summary of initial strain input for different element types.

Table 11.6-1: Initial strains input for nodes and elements

Element	Nodal point strain components						Element strain components					
	e_{11}	e_{22}	e_{33}	γ_{12}	γ_{13}	γ_{23}	e_{11}	e_{22}	e_{33}	γ_{12}	γ_{13}	γ_{23}
Truss	✓	-	-	-	-	-	✓	-	-	-	-	-
2-D solid	✓	✓	✓ ²⁾	✓	-	-	✓ ³⁾	✓ ³⁾	✓ ^{2,3)}	-	-	-
3-D solid	✓	✓	✓	✓	✓	✓	✓ ³⁾	✓ ³⁾	✓ ³⁾	-	-	-
Beam	✓	-	-	-	-	-	✓	-	-	-	-	-
Iso-beam	✓	-	✓ ⁴⁾	-	-	-	✓	-	✓ ⁴⁾	-	-	-
Plate:												
membrane,	✓	✓	-	✓	-	-	✓	✓	-	✓	-	-
flexural												
Shell:												
membrane,	✓	✓	-	✓	✓	✓	✓	✓	-	✓	✓	✓
gradient												
Pipe	✓	-	-	-	-	-	✓	-	-	-	-	-

- 1) The symbol "✓" means that the strain component can be defined for that element type, either by node or by element input.
- 2) Axisymmetric and generalized plane strain elements only.
- 3) Entered in the global system or 2D local system using strain field coefficients.
- 4) Axisymmetric shell elements only.

Initial strains for large displacement / small strain formulation

For the large displacement / small strain formulation, the initial strains are interpreted as initial Green-Lagrange strains, the initial strains are added to the Green-Lagrange strains computed from the displacements, and the result is used to compute the 2nd Piola-Kirchhoff stresses, that is

$${}_0\mathbf{S} = {}_0\mathbf{S}({}_0\boldsymbol{\varepsilon} + {}^0_0\boldsymbol{\varepsilon})$$

This calculation is justified only when the strains (both total and initial) are small.

Initial strains in large strain formulations, 2D and 3D elements:

The material models discussed in this section are the large strain plasticity models (in which the ULH formulation is used) and also the rubber-like material models.

The initial strains are assumed to be small, and the initial spin tensor is assumed to be equal to zero, so that

$${}^t_0\mathbf{X} = (\mathbf{I} + {}^0\mathbf{e})^{-1}$$

in which ${}^0\mathbf{e}$ is the initial strain tensor and ${}^t_0\mathbf{X}$ is the initial deformation gradient. Then a multiplicative decomposition of the deformation gradient is used at every step, i.e.:

$${}^t_0\mathbf{X} = {}^t_0\mathbf{X}({}^t_0\mathbf{X})^{-1}$$

where ${}^t_0\mathbf{X}$ is the total deformation gradient (computed from the element displacements at time t) and ${}^t_0\mathbf{X}$ is the mechanical deformation gradient (used to compute the stresses). Note that with the above convention, tensile initial strains applied to an element that is unrestrained cause the element to shrink, as in Fig 11.6-1. This follows because an unrestrained element has zero stresses, hence ${}^t_0\mathbf{X} = \mathbf{I}$, ${}^t_0\mathbf{X} = {}^t_0\mathbf{X}({}^t_0\mathbf{X})^{-1}$ and $\det {}^t_0\mathbf{X} < 1$ (for tensile initial strains).

- The components of ${}^t_0\mathbf{X}$ are output when deformation gradients are requested to be output.

Initial strains in inelastic materials

We briefly discuss initial strains in inelastic materials, such as plasticity, creep and viscoelastic materials. For simplicity, we will assume that the formulation is materially-nonlinear-only, however similar considerations arise in large displacement and large strain analysis.

The rule $\boldsymbol{\tau} = \boldsymbol{\tau}(\mathbf{e} + {}^0\mathbf{e})$ is used for inelastic materials, where the

function $\boldsymbol{\tau}()$ includes inelastic variables such as the plastic strain as well as the strain $\mathbf{e} + {}^0\mathbf{e}$.

One subtle point is that this rule is also applied at the very beginning of the analysis, namely to compute the stresses at time TSTART. At the beginning of the analysis, $\mathbf{e} = \mathbf{0}$, so the stresses at time TSTART are computed using ${}^0\boldsymbol{\tau} = \boldsymbol{\tau}({}^0\mathbf{e})$. These stresses need not be in equilibrium with the applied forces, since equilibrium is not computed at time TSTART.

In addition, if the material is rate-dependent, the calculation of ${}^0\boldsymbol{\tau} = \boldsymbol{\tau}({}^0\mathbf{e})$ includes the strain rate. The strain rate also includes effects due to initial strains, and the strain rate depends on the time step size. The time step size used in the first time step is also used in the calculation of ${}^0\boldsymbol{\tau} = \boldsymbol{\tau}({}^0\mathbf{e})$. This can occasionally cause difficulties, for example in explicit analysis, in which the first time step is very small, and therefore in which the strain rate as computed from the initial strain is large.

11.6.5 Initial stresses

- An initial stress option can be used with all element types. The following material models can be used: **elastic-isotropic, elastic-orthotropic, thermo-elastic, thermo-orthotropic, Drucker-Prager, Cam-clay, Mohr-Coulomb**.

The initial stress capability allows initial stresses corresponding to an unknown loading to be input. These stresses can be provided as nodal point initial stresses or element initial stresses.

Whenever the constitutive relation is used, the initial stresses are added to the current stresses caused by the strains. The initial stresses are included in the printed and saved stresses.

- Either initial stresses or initial strains can be accepted by the program. However the choice of initial stresses or initial strains is a program-wide choice, so initial stresses and initial strains cannot be used together in the same analysis.
- The following equations are used:

$$\boldsymbol{\tau} = \boldsymbol{\tau}^M + \boldsymbol{\tau}^I \quad (11.6-1)$$

$$\mathbf{F} = \int_V \mathbf{B}^T \boldsymbol{\tau} dV \quad (11.6-2)$$

in which $\boldsymbol{\tau}^I$ are the initial stresses (corresponding to the formulation/material model used, i.e. Cauchy or 2nd Piola-Kirchhoff), and $\boldsymbol{\tau}^M$ are the stresses calculated as usual from the mechanical strains. Equation (11.6-2) yields at time 0

$$\mathbf{R}^I = \mathbf{F}^I = \int_V \mathbf{B}^T \boldsymbol{\tau}^I dV$$

- Please note:
 - ▶ The stresses $\boldsymbol{\tau}^I$ are in the \mathbf{K}_{NL} matrix (stiffening effect)
 - ▶ The stresses $\boldsymbol{\tau}^I$ are added to the stresses computed in the material law.
 - ▶ The stresses $\boldsymbol{\tau}^I$ are effects that do not change in time.
- There are two options for the initial stress feature:

Initial stresses do not cause deformations

The step-by-step incrementation is performed using

$${}^{t+\Delta t}\mathbf{K}^{(i-1)}\Delta\mathbf{U}^{(i)} = {}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F}^{(i-1)} + \mathbf{R}^I$$

in which ${}^{t+\Delta t}\mathbf{R}$ is the externally applied load vector. Use the AUI command `MASTER INITIALSTRESS=YES` to request this option.

Figure 11.6-2 shows a single 2-D element in which this option is used, and in which no other applied loads or body forces are present. Even though the stresses are non-zero, no deformations occur, because \mathbf{R}^I is added to the out-of-balance force vector. This modeling technique can be very useful in the analysis of geologic materials.

This option can be used in linear and materially-only-nonlinear analysis.

However, it should be noted that if the birth/death element option is used (e.g. tunneling), then the initial stresses cause deformations.

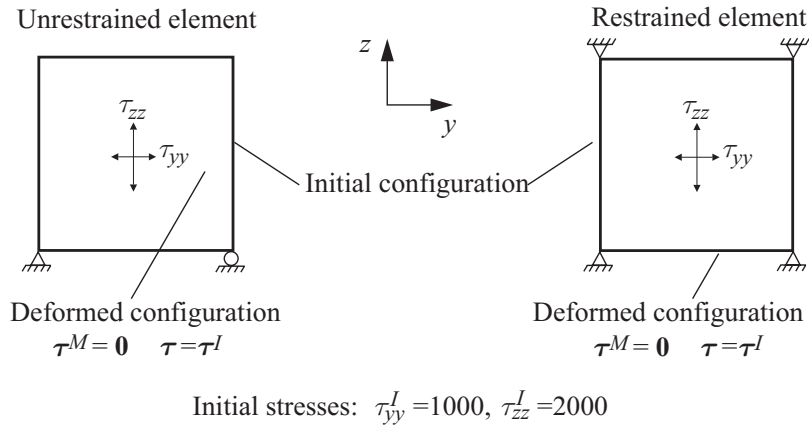


Figure 11.6-2: Initial stresses applied to 2-D solid element, initial stresses do not cause deformations

Initial stresses cause deformations

The step-by-step incrementation is performed using

$${}^{t+\Delta t}\mathbf{K}^{(i-1)}\Delta\mathbf{U}^{(i)} = {}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F}^{(i-1)}$$

Use the AUI command MASTER INITIALSTRESS=DEFORMATION to request this option.

Figure 11.6-3 shows a single 2-D element in which this option is used, and in which no other applied loads or body forces are present. Tensile initial stresses are applied. When the element is unrestrained, the stresses are zero, and the element shrinks because the mechanical stresses are compressive. When the element is rigidly restrained from motion, $\tau^M = \mathbf{0}$, hence the element contains tensile stresses.

This option can be used in linear, materially-only-nonlinear analysis and large displacement/small strain analysis.

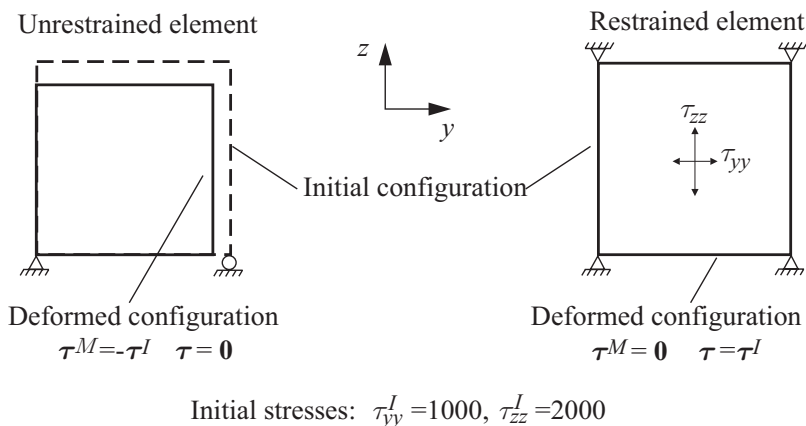


Figure 11.6-3: Initial stresses applied to 2-D solid element, initial stresses cause deformations

Initial stress input

- When initial stresses are chosen, then all initial stress input is entered using the corresponding initial strain input commands.

Initial stress components are interpreted in an initial strain axes coordinate system.

The initial stress components can be entered either using nodal point initial stresses or element initial stresses. Simply substitute the initial stress for initial strain in the commands that define initial strains. For example, to assign membrane element initial stresses to shell elements, the commands

```
* SHELL ELEMENT
EDATA
ENTRIES EL    EPS11    EPS22    EPS12    EPS13    EPS23
          1  1000.0   2000.0   500.0   200.0   -100.0
DATAEND
```

could be used. Table 11.6-1 applies to initial stresses, provided that the stress components are substituted for the strain components.

Restrictions

- The initial stress option can be used for the Drucker-Prager,

Cam-clay and Mohr-Coulomb material models only if the effective stress corresponding to the initial stresses is smaller than the yield stress (that is, the initial stress state must lie within the initial yield surface).

- The initial stress option should not be used in large displacement/large strain analysis.

11.7 Energy calculation

11.7.1 Energy printout

- The Energy View utility enables users to monitor energy variations of the whole model or specific parts of the model during analysis; it also allows users to load and review the energy data following analysis. The energy curves can be displayed graphically in the Energy View dialog box and/or be saved to an energy (.egy) file.
- ADINA supports Energy View for 2D/3D solid elements, beam elements, iso beam elements, truss elements, spring elements, 4-node shell elements, 3D-shell elements and 2D/3D contact.
- Energy data are calculated and saved either at the level of whole model or at the element group level, including contact group, concentrated mass and concentrated damper. The `ENERGY` command is used to specify which parts of the model should be included in the energy data calculation. The `ENERGY` command accepts four options: `NO`, `WHOLE`, `EG-ALL` and `EG-LIST`.
 - ▶ `NO`: no energy data are saved
 - ▶ `WHOLE`: energy data of the whole model are saved
 - ▶ `EG-ALL`: energies of all element groups are saved
 - ▶ `EG-LIST`: energies of selected element groups are saved
- Note that except the option `NO`, the energy data for the whole model is always saved. For the last two options, the energy data for contact groups, concentrated masses, and concentrated dampers are always saved.

- The NODESAVE-STEPS command is used to specify the time steps for which energy results are to be saved.

Energy calculation for element group

The following energy items are calculated for element groups:

- | | |
|---|-------|
| ► Elastic strain energy | E_E |
| ► Inelastic (viscoplastic) dissipations | E_P |
| ► Kinetic energy | E_K |
| ► Thermal strain energy | E_T |
| ► Creep strain energy | E_C |
| ► Visco strain energy | E_V |

- The equations used to calculate element strain energies are:

Linear formulation:

$$E_E = \int_V \frac{1}{2} \tau_{ij} \gamma_{ij} dV \quad (11.7-1)$$

$$E_T = \int_V \frac{1}{2} \tau_{ij} \gamma_{ij}^T dV \quad (11.7-2)$$

MNO formulation:

$$E_E = \int_V \int_t \frac{1}{2} \left({}^t\tau_{ij} + {}^{t+\Delta t}\tau_{ij} \right) d\gamma_{ij}^e dV \quad (11.7-3)$$

$$E_P = \int_V \int_t \frac{1}{2} \left({}^t\tau_{ij} + {}^{t+\Delta t}\tau_{ij} \right) d\gamma_{ij}^p dV \quad (11.7-4)$$

$$E_T = \int_V \int_t \frac{1}{2} \left({}^t\tau_{ij} + {}^{t+\Delta t}\tau_{ij} \right) d\gamma_{ij}^T dV \quad (11.7-5)$$

$$E_C = \int_V \int_t \frac{1}{2} \left({}^t\tau_{ij} + {}^{t+\Delta t}\tau_{ij} \right) d\gamma_{ij}^C dV \quad (11.7-6)$$

TL formulation:

for linear materials

$$E_E = \int_V \frac{1}{2} S_{ij} \varepsilon_{ij} dV \quad (11.7-7)$$

$$E_T = \int_V \frac{1}{2} S_{ij} \varepsilon_{ij}^T dV \quad (11.7-8)$$

for plastic materials

$$E_E = \int_V \int_t \frac{1}{2} ({}^t S_{ij} + {}^{t+\Delta t} S_{ij}) d\varepsilon_{ij}^e dV \quad (11.7-9)$$

$$E_P = \int_V \int_t \frac{1}{2} ({}^t S_{ij} + {}^{t+\Delta t} S_{ij}) d\varepsilon_{ij}^p dV \quad (11.7-10)$$

$$E_T = \int_V \int_t \frac{1}{2} ({}^t S_{ij} + {}^{t+\Delta t} S_{ij}) d\varepsilon_{ij}^T dV \quad (11.7-11)$$

$$E_C = \int_V \int_t \frac{1}{2} ({}^t S_{ij} + {}^{t+\Delta t} S_{ij}) d\varepsilon_{ij}^C dV \quad (11.7-12)$$

ULH formulation:

$$E_E = \int_V \int_0^t \frac{1}{2} (\bar{\tau}_{ij} + {}^{t+\Delta t} \bar{\tau}_{ij}) dE_{ij}^e dV \quad (11.7-13)$$

$$E_P = \int_V \int_0^t \frac{1}{2} ({}^t \bar{\tau}_{ij} + {}^{t+\Delta t} \bar{\tau}_{ij}) dE_{ij}^p dV \quad (11.7-14)$$

$$E_T = \int_V \int_0^t \frac{1}{2} (\bar{\tau}_{ij} + {}^{t+\Delta t} \bar{\tau}_{ij}) dE_{ij}^T dV \quad (11.7-15)$$

$$E_C = \int_V \int_0^t \frac{1}{2} (\bar{\tau}_{ij} + {}^{t+\Delta t} \bar{\tau}_{ij}) dE_{ij}^C dV \quad (11.7-16)$$

ULJ formulation:

$$E_E = \int_V \int_0^t \frac{1}{2} ({}^t \bar{\tau}_{ij} + {}^{t+\Delta t} \bar{\tau}_{ij}) dD_{ij}^e dV \quad (11.7-17)$$

$$E_P = \int_V \int_0^t \frac{1}{2} \left({}^t\bar{\tau}_{ij} + {}^{t+\Delta t}\bar{\tau}_{ij} \right) dD_{ij}^P dV \quad (11.7-18)$$

$$E_T = \int_V \int_0^t \frac{1}{2} \left({}^t\bar{\tau}_{ij} + {}^{t+\Delta t}\bar{\tau}_{ij} \right) dD_{ij}^T dV \quad (11.7-19)$$

$$E_C = \int_V \int_0^t \frac{1}{2} \left({}^t\bar{\tau}_{ij} + {}^{t+\Delta t}\bar{\tau}_{ij} \right) dD_{ij}^C dV \quad (11.7-20)$$

The visco strain energy is calculated for three-network material model:

$$E_V = \int_V \mathcal{D}_{\text{int}} dV$$

In the above equations:

τ_{ij} : Cauchy stress

γ_{ij} : Engineering strain

S_{ij} : 2nd P-K stress

ε_{ij} : Green strain

$\bar{\tau}_{ij}$: Rotated Cauchy stress

E_{ij} : Right Hencky strain

D_{ij} : Jaumann strain

\mathcal{D}_{int} : The internal energy dissipation of three-network material
(Eq. 3.17-24)

superscript e : elastic part

superscript p : plastic part

superscript T : thermal part

superscript C : creep part

The element kinetic energy is

$$E_K = \frac{1}{2} \dot{\mathbf{U}}^T \mathbf{M} \dot{\mathbf{U}} \quad (11.7-21)$$

where

$\dot{\mathbf{U}}$: element velocity vector

\mathbf{M} : element mass matrix

Energy calculations for contact groups

The following energy items are calculated for contact groups

- ▶ Contact frictional energy E_F
- ▶ Contact compliance energy stored E_C
- ▶ Contact damping energy dissipated E_{CD}

The equations for these quantities are:

$${}^{t+\Delta t}E_F = {}^tE_F + \frac{1}{2} \left({}^t\mathbf{F}_C + {}^{t+\Delta t}\mathbf{F}_C \right) \cdot \Delta \mathbf{U}_C^T + \frac{1}{2} \left({}^t\mathbf{F}_T + {}^{t+\Delta t}\mathbf{F}_T \right) \cdot \Delta \mathbf{U}_T^T, \quad (11.7-22)$$

$${}^{t+\Delta t}E_C = {}^tE_C + \frac{1}{2} \left({}^t\mathbf{F}_C + {}^{t+\Delta t}\mathbf{F}_C \right) \cdot \Delta \mathbf{U}_C^N + \frac{1}{2} \left({}^t\mathbf{F}_T + {}^{t+\Delta t}\mathbf{F}_T \right) \cdot \Delta \mathbf{U}_T^N, \quad (11.7-23)$$

and

$${}^{t+\Delta t}E_{CD} = {}^tE_{CD} + \frac{1}{2} \left({}^t\mathbf{F}_{DC} + {}^{t+\Delta t}\mathbf{F}_{DC} \right) \cdot \Delta \mathbf{U}_C + \frac{1}{2} \left({}^t\mathbf{F}_{DT} + {}^{t+\Delta t}\mathbf{F}_{DT} \right) \cdot \Delta \mathbf{U}_T. \quad (11.7-24)$$

In the above equations:

\mathbf{F}_C : Contact force on contactor nodes

\mathbf{F}_T : Contact force distributed on target nodes

$\Delta \mathbf{U}_C^T$: Incremental displacement of contactor nodes along tangential direction

$\Delta \mathbf{U}_T^T$: Incremental displacement of target nodes along tangential direction`

$\Delta \mathbf{U}_C^N$: Incremental displacement of contactor nodes along normal direction

$\Delta \mathbf{U}_T^N$: Incremental displacement of target nodes along normal direction

\mathbf{F}_{DC} : Contact damping force on contactor nodes

\mathbf{F}_{DT} : Contact damping force on target nodes

$\Delta \mathbf{U}_C$: Incremental displacement of contactor nodes

$\Delta \mathbf{U}_T$: Incremental displacement of target nodes

Energy calculation for concentrated mass and concentrated damper

The kinetic energy for concentrated masses is calculated using equation 11.7-11, but with the velocity vector of the whole model and mass matrix associated with the concentrated mass.

The Rayleigh damping energy of concentrated dampers is

$${}^{t+\Delta t}E_D = {}^tE_D + \frac{1}{2}({}^t\mathbf{C}^t\dot{\mathbf{U}} + {}^{t+\Delta t}\mathbf{C}^{t+\Delta t}\dot{\mathbf{U}})({}^{t+\Delta t}\mathbf{U} - {}^t\mathbf{U}) \quad (11.7-25)$$

where

$\dot{\mathbf{U}}$: velocity vector of the whole model

\mathbf{U} : displacement vector of the whole model

\mathbf{C} : Rayleigh damping matrix associated with the concentrated

dampers

Energy calculation for whole model

The following energy items are calculated for the whole model:

▶ Initial energy	0E
▶ Work done by external force	W_{Ex}
▶ Elastic strain energy stored in the model	E_E
▶ Plastic strain energy dissipated	E_P
▶ Kinetic energy of the current time	E_K
▶ Rayleigh damping energy dissipated	E_D
▶ Visco strain energy dissipated	E_V
▶ Thermal strain energy stored	E_T
▶ Creep strain energy dissipated	E_C
▶ Contact frictional energy dissipated	E_F
▶ Contact compliance energy stored	E_{CC}
▶ Contact damping energy dissipated	E_{CD}
▶ Energy balance	E_B

The initial energy 0E is an internal variable calculated in ADINA, but it is never saved for Energy View.

The external work is

$${}^{t+\Delta t}W_{Ex} = {}^tW_{Ex} + \frac{1}{2}({}^t\mathbf{R} + {}^{t+\Delta t}\mathbf{R})({}^{t+\Delta t}\mathbf{U} - {}^t\mathbf{U}), \quad (11.7-26)$$

\mathbf{R} : global loading vector

\mathbf{U} : global displacement vector

Strain energy E_E , E_P , E_V , E_T and E_C , contact energies E_F , E_{CC} and E_{CD} for the whole model are calculated by summing the contributions of each element group or contact group.

The kinetic energy and damping energy of the whole model are calculated using equations 11.7-21 and 11.7-25, but with the mass and damping matrices of the whole model.

The energy balance is calculated using the following equation, reflecting the balance between the energies input to the system

(initial energy and external work) and energies stored in the system or dissipated.

$$E_B = \underbrace{({}^0E)}_{\text{Initial Energy}} + \underbrace{(W_{Ex})}_{\text{Energy In}} - \underbrace{(E_K + E_E + E_C)}_{\text{Energy Stored}} - \underbrace{(E_P + E_D + E_F + E_{CD})}_{\text{Energy Dissipated}} \quad (11.7-17)$$

$$E_B = \underbrace{({}^0E)}_{\text{Initial Energy}} + \underbrace{(W_{Ex})}_{\text{Energy In}} - \underbrace{(E_K + E_E + E_C)}_{\text{Energy Stored}} - \underbrace{(E_P + E_D + E_F + E_{CD})}_{\text{Energy Dissipated}}$$

The initial energy is always calculated when there's initial strain, initial velocity or bolt loading. For analysis with initial stress, the initial energy is calculated only for plastic (bilinear, multi-linear, orthotropic, creep, etc.) material. A special case is that when there's only initial velocity, but no initial strain/stress, the initial kinetic energy is only calculated for total energy. The element kinetic energy cannot be calculated in this case.

The energy results at step zero are only printed when there's initial energy. Otherwise we don't print the energy results at step zero.

For analysis with one time step solution without initial energy, the energy results are plotted as points rather than curves in the Energy View.

Currently, ENERGY command doesn't support the following types of element and material:

- ▶ Pipe element
- ▶ Electric-related energy in piezoelectric material
- ▶ Mullin effect in rubber-like material
- ▶ Visco effect in visco-elastic material (Visco effect in three-network material is supported)

The energy balance should be close to zero. If greater than zero, energy is being lost, whereas if smaller than zero, energy is gained. A non-zero energy balance might indicate an inaccurate solution, for example, due to spurious numerical energy dissipation from a time step size that is too large. Alternatively, a non-zero energy

balance might be due to a source of energy gain or lost from a element or material type not supported by ENERGY command.

11.7.2 Strain energy densities

- The computation of strain energy densities at integration points can be requested for 2-D solid and 3-D solid elements.
- The strain energy density can be requested for all material models, and all nonlinear formulations. The mixed (u/p) formulation can be used.
- The strain energy density cannot be requested in explicit time integration analysis.
- The strain energy density calculations used in this section are also used for fracture mechanics (Chapter 10).
- The strain energy density for rubber-like materials is computed directly from the definitions given in Section 3.8.
- The remainder of this section describes the calculation of the strain energy density for the other material models.

Small displacements / small strains (linear and MNO formulations)

- The basic formula for the strain energy density under small displacement and small strain conditions is

$$dW = \tau_{ij} de_{ij} \quad (11.7-18)$$

where τ_{ij} is the Cauchy stress tensor and e_{ij} is the small strain tensor.

- When there are thermal strains, the following formula is used:

$$dW = \tau_{ij} de_{ij}^m \quad (11.7-19)$$

in which $e_{ij}^m = e_{ij} - e_{ij}^{th}$ are the mechanical strains.

Large displacements / small strains (TL formulation)

- The basic formula for the strain energy density under large displacement and small strain conditions is

$$d {}^t_0 W = {}^t_0 S_{ij} d {}^t_0 \varepsilon_{ij} \quad (11.7-20)$$

where the left subscript means “referred to the original configuration, or “per unit original volume”, ${}_0^t S_{ij}$ are the 2nd Piola-Kirchhoff stresses and ${}_0^t \varepsilon_{ij}$ are the Green-Lagrange strains. When there are thermal effects, the thermal strains are subtracted from ${}_0^t \varepsilon_{ij}$, just as in the small displacement / small strain case.

Large displacements / large strains (ULH formulation)

- The basic formula for the strain energy density under large displacement and large strain conditions, using the ULH formulation, is

$$d {}^t_0 W = {}^t \overline{\tau}_{ij} d {}^t_0 E_{ij}^R \quad (11.7-21)$$

where ${}^t \overline{\tau}_{ij}$ is the rotated Cauchy stress and ${}_0^t E_{ij}^R$ is the right Hencky strain. Both of these quantities are discussed in Section 3.1.2. Notice that although the rotated Jaumann stress should theoretically be used, the rotated Cauchy stress is used instead, since the plastic strains are incompressible. When there are thermal effects, the thermal strains are subtracted from ${}_0^t E_{ij}^R$, just as in the small displacement / small strain case.

Large displacements / large strains (ULJ formulation)

- The basic formula for the strain energy density under large displacement and large strain conditions, using the ULJ formulation, is

$$d {}^t_0 W = {}^t \tau_{ij} D_{ij} \quad (11.7-22)$$

where D_{ij} is the rate of deformation tensor. As in the ULH case, the Jaumann stress should theoretically be used, but the Cauchy stress is used instead. When there are thermal effects, the rate of change of thermal strains is subtracted from D_{ij} .

Numerical evaluation

- The basic formula $dW = \tau_{ij} de_{ij}$ is numerically integrated using the trapezoidal rule, e.g.

$${}^{t+\Delta t}W = {}^tW + \frac{1}{2}({}^t\tau_{ij} + {}^{t+\Delta t}\tau_{ij})({}^{t+\Delta t}e_{ij} - {}^te_{ij}) \quad (11.7-23)$$

in which ${}^0W = 0$ is assumed. Similar expressions are used for the large displacement formulations.

- The accuracy of the strain energy density depends on the time step size, when there are nonlinearities in the element group.

11.8 Element group inertial properties

- The following element group properties can be requested in ADINA for printout/saving: **total mass, total volume, moments of inertia, products of inertia, centroid, center of mass.**
- General elements and spring elements do not contribute to the element group inertial properties.
- The element group inertial properties are computed at the beginning of the analysis based on the undeformed model.
- In ADINA, the moments and products of inertia are calculated with respect to the origin of the global coordinate system.
- For axisymmetric element groups, the volume, mass and other properties are calculated using a thickness of 2π radians. Exception: the masses in the three coordinate directions are computed using a thickness of 1 radian; this facilitates comparison

with the modal masses.

- The AUI can combine the element group quantities over the whole model or over a selected zone in the model. The AUI also includes the concentrated masses in the calculations. The moments and products of inertia are calculated and printed with respect to the center of mass of the whole model or zone.
- When the concentrated masses have different values in different coordinate directions, for example, when the concentrated mass in the x direction has a different value than the concentrated mass in the y direction, then the AUI uses the following procedure: for each such concentrated mass, the AUI computes an 'average' mass as the average of the nonzero masses for the different directions. The AUI then computes the total mass as the sum of the element group masses plus the sum of the average concentrated masses. In the listing of mass properties, the AUI lists the total mass, and also the sum of the mass values in the different directions (without averaging). For the center of mass and the inertial properties, the AUI uses the total mass (with averaging) throughout.
- When listing the percent modal masses (MPFINFO command), the AUI uses the total mass values in the different directions (without averaging the concentrated masses in the different directions).
- It is not possible for the AUI to compute the above-mentioned properties for a selected part of an element group.

11.9 Restart

- Restart is a useful feature in the ADINA System. It can be used when the user wishes to continue an analysis beyond its previous end point, or change the analysis type, loads or boundary conditions or tolerances. A restart analysis is set via the MASTER command. Recovery of results from a restart file without continuing the analysis is also possible.
- All relevant solution data needed for a restart run are saved in a file (extension .res) in case they are needed in a restart analysis. The frequency of data writing to a restart file is set via the

MASTER command.

- Note that multiple restart data can be appended to the restart file. Also, the solution times for which the restart file has results can be specified. This enables the restart analysis to be based on a solution step different from the last converged solution. However, saving multiple time step solutions to a restart file can lead to a large restart file size.
- To perform a restart analysis, the .res file from the first model must be copied to the second model. When running in interactive mode, the AUI will first look for the restart file, and if it is not found, the user will be prompted to locate the restart file. When running in batch mode the .res file must be copied before starting the second analysis.
- If no restart time is provided in the second model (achieved by setting the restart from 0.0), the program uses the data for the latest restart time on the .res file.
- Note that once the second analysis starts, it will overwrite the .res file with new data. Therefore, if the user wishes to redo the second run, then the .res file must be copied again from the first model. This does not apply to when restart data is appended to the .res file and when the second run is a frequency analysis (no restart data in this case).
- The geometry, and most element data cannot be changed in a restart analysis. However, the following changes are allowed:
 - ▶ Type of analysis can change. Static to dynamic and dynamic to static restarts are allowed.
 - ▶ Solution type can change. Implicit analysis to explicit analysis restarts are allowed and vice-versa. In this case, features not available in either solution type cannot be used. Note that some default settings are different between implicit and explicit analysis. Some of these have to be manually set by the user to enable restarts. The most common setting is:
 - Default large strain formulation (ULH in implicit and ULJ in explicit)

- ▶ Solution control variables can change. The flags, constants and tolerances for the iteration method, convergence, time integrations, automatic time stepping and load-displacement-control can be changed.
 - ▶ Externally applied loads and enforced displacements can be modified.
 - ▶ The material constants can be changed. However, note that in a restart run the same material model (with the same number of stress-strain points and the same number of temperature points, if applicable) must be used for each element as in the preceding run.
 - ▶ Boundary conditions can change.
 - ▶ Constraint equations and rigid elements can change.
 - ▶ Contact settings can change. This includes most contact group, contact pair and contact surface parameters.
 - ▶ Rayleigh damping coefficients can change.
 - ▶ Time increment and number of solution steps can be modified.
 - ▶ Time functions describing the load variations can be changed.
 - ▶ It is not allowed to have no birth-death in the first run, then birth-death in the restart run. If there is no birth-death in the first run, give a large value for TDEATH for the element groups that have birth-death in the restart run.
- When restarting from static to dynamic analysis (both implicit and explicit dynamics), the initial velocities and accelerations are assumed to be zero. However, if an initial velocity is prescribed in the restart run, it will be used instead. When restarting from one dynamic analysis to another, initial velocities and accelerations are transferred from the first to the second run.
 - When changing from implicit to explicit analysis (or from explicit to implicit analysis), be aware that the AUI defaults depend on the analysis type. For example, by default the AUI selects the ULH formulation in implicit analysis and selects the ULJ

formulation in explicit analysis.

- The RESTART-INFO command can be used to obtain detailed information from a previously created restart file such as: the date the restart file was written, the name of the data file used to create the restart file, the solution times for which results are saved to the restart file, and information regarding the model saved to the restart file (number of nodes, number of elements, number of element groups, etc.).

Result recovery

In result recovery mode, ADINA reads a restart file and generates results for the last saved restart step. No analysis is performed. If multiple restarts are saved in the restart file, results can be recovered at any of the saved times by setting the restart time.

Restart in frequency analysis

- For large systems, if the subspace iteration method for frequency analysis does not converge, it can be efficient to use the converged eigenvectors as a subset of the starting vectors and redo the analysis with modified input. The advantage is that the amount of computational effort needed to compute the remaining eigenvectors is less than the amount of computational effort needed to compute all of the requested eigenvectors.
- In the following discussion, the first ADINA run is the one in which not all of the requested eigenvectors are obtained and the second ADINA run is the one in which the remaining requested eigenvectors are obtained. Note that the second ADINA run is not considered to be a restart run.

You supply the converged eigenvectors from the first ADINA run to the second ADINA run as user-supplied starting vectors. You also generate a new data file for the second ADINA run, in which you can change the number of starting vectors and the maximum number of iterations. It is recommended that the number of starting vectors be greater than or equal to the number of converged vectors from the first ADINA run.

Before starting the second ADINA run, you need to copy the eigenvectors from the first run to the second run. The eigenvectors

are stored in a file with extension .stv.

- Restart can also be used to switch from one analysis type to another. The following table provides details of the possible analysis type switches.

Table 11.9-1: Switching between analysis options with restart

<div>From \ To</div>	Statics	Implicit Dynamics	Explicit Dynamics	Frequency Analysis	Modal Analysis
Statics	✓	✓	✓	✓	x ⁽¹⁾
Implicit Dynamics	✓	✓	✓	✓	x ⁽¹⁾
Explicit Dynamics	✓	✓	✓	✓	x ⁽¹⁾
Frequency Analysis	✓	✓	✓	x	✓ ⁽²⁾
Modal Analysis	✓	✓	✓	x	x

Notes:

(1) Must restart to frequency analysis first

(2) Can also be in the same run (no restart needed)

11.10 Parallel processing

- ADINA supports shared memory parallel processing (SMP) and distributed memory parallel processing (DMP).
- For shared memory parallel processing the following coding is parallelized: assembly of the global system matrices (loop over element groups), the in-core sparse solver and the out-of-core sparse solver. The parallelization of the assembly of the global system matrices is available for all platforms except for 32-bit Windows.
- To benefit from parallel element assembly, element groups must be divided into subgroups. The number of subgroups should be equal or greater to the number of processors (preferably a multiple of the number of processors). This is set via the NSUBG and MINEL parameters in the EGCONTROL command.
- For distributed memory parallel processing there are currently two options:
 - ▶ For static and implicit dynamic solutions, the sparse solver is parallelized using MPI directives, and matrix assembly using shared memory OpenMP directives. Therefore we have a combined DMP-SMP algorithm to run ADINA effectively.
 - ▶ For explicit time integration, the program divides models into subdomains and uses MPI directives to obtain solutions. Shared memory OpenMP directives are also used in local subdomains.

11.11 Usage of memory and disk storage

- Depending on the size of the problem and the memory allocated to ADINA (MTOT), ADINA can perform the solution either in-core (entirely within real or virtual memory) or out-of-core (reading and writing disk files). Whenever possible the solution is performed in-core.
- Explicit dynamic analysis can only run in-core. Enough memory must be available.
- Depending on the size of the problem and the available memory, there are 2 internal in-core/out-of-core settings for the program without the solver. These 2 memory settings are:
 - ▶ The stiffness (and mass) matrices and the element information are all stored in-core. (IOPTIM=3)
 - ▶ The stiffness (and mass) matrices are stored in-core, while the element information is kept out-of-core. (IOPTIM=2)
- The memory setting used by the program is provided in the .out file as parameter variable IOPTIM.
- The sparse solver and the 3D-iterative solver may both be set to in-core or out-of-core.
- When using the iterative multi-grid solver (Section 6.5.2), the solution is always performed in-core. The out-of-core solution procedure would take an unreasonably long time in most cases.
- The disk files that may become large are those with the following file name extensions:
 - 1*: Linear element group data
 - 2*: Nonlinear element group data
 - 4*: Linear stiffness matrix
 - 7*: Implicit dynamic analysis, effective stiffness matrix
 - 8*: Restart file
 - 11*: Mass/damping matrices

11.12 Miscellaneous

- The program can write intermediate files to disk for external processing, and can read various inputs (such as stiffness and mass matrices) directly from a user-provided file (such as pipe pressures, forces, or displacements).
- **Saving stiffness and mass/damping matrices using disk files**

The DISK-STORAGE command can be used to write matrices on the *.mtx disk file:

```
DISK-STORAGE ...GLOBAL-MATRIX=NONE/SAVE/NODE  
/FULL STIFFNESS-MATRIX=NONE/SAVE MASS-MATRIX  
=NONE/SAVE DAMPING-MATRIX=NONE/SAVE
```

For a detailed description of the DISK-STORAGE command, see the ADINA Command Reference Manual.

A non-factorized, linear or nonlinear, global stiffness matrix can be saved using a disk file (*.mtx file), for a given time step. For implicit dynamic solutions a mass matrix (lumped or consistent) and Rayleigh damping matrix can also be stored on the same *.mtx file if requested. There are three available options:

Option 1. Command input: GLOBAL-MATRIX=FULL

Matrices are given in full upper triangular by column heights form. The program will then write stiffness, mass and damping matrices for the last solution time only. The *.mtx file will contain the following data:

- MAXA array for full by column heights stiffness/mass/damping matrices
- Full by column heights stiffness matrix followed by mass and damping matrices (if required)

Option 2. Command input: GLOBAL-MATRIX=SAVE

The matrices are given in a “sparse” form, i.e. zero matrix elements in the columns are removed and can be written to the *.mtx file for any time step. The *.mtx file always contains the following data:

- MAXA array, for now column heights vector with the zero elements in the columns removed (from the matrix diagonal for symmetric matrices, from the matrix bottom for non-symmetric

matrices)

- Equation numbers for the nonzero matrix elements
- Stiffness matrix followed by mass and damping matrices (if requested) according to the MAXA array used here

Option 3. Command input: GLOBAL-MATRIX=NODE

The matrices (i.e. stiffness, mass and damping) are given in a sparse form and can be written out to the *.mtx file for any solution time step. The matrices are written for every node in a “direct matrix input” format required by the AUI. The matrices can then be used in other models as input data for the direct matrix input options.

- **Saving heat conductivity and heat capacity matrices**

Heat conductivity and heat capacity matrices can also be stored on the *.mtx file (see command DISK-STORAGE for details). However, only Option 1 and 2 are available for heat conductivity and heat capacity matrices.

- In certain cases, you can specify data to ADINA using disk files:

Pipe internal pressures: Pipe internal pressures can be read from unit 24 in the following format (unformatted sequential Fortran file):

```
READ (24) TIME, (PIPEIP(I), I=1, NUMNP)
```

Here NUMNP is the number of nodes in the main structure and TIME is the solution time for the pipe internal pressures.

You must specify whether the times in unit 24 correspond exactly with the ADINA solution times, or whether they are different than the ADINA solution times. In the latter case, ADINA interpolates the pipe internal pressures to the ADINA solution times and writes the interpolated data to unit 24.

Temperatures: Temperatures can be read from unit 56 in a manner entirely analogous to the reading of pipe internal pressures. Temperatures read from unit 56 are added to the other specified temperatures (initial or prescribed), except at the beginning of a restart run.

Temperature gradients: Temperature gradients can be read from unit 57 in a manner entirely analogous to the reading of temperatures. Temperature gradients read from unit 57 are added to the other specified temperature gradients (initial or prescribed), except at the beginning of a restart run.

Forces and displacements: Forces can be read from unit 58 in the following format (unformatted sequential Fortran file):
`READ (58) TIME, ((F(I,J), I=1, NUMNP), J=1, NOFC)`
Here NOFC is the number of force components. You must specify which force component corresponds to each value of J.

You must specify whether the times in unit 58 correspond exactly with the ADINA solution times, or whether they are different than the ADINA solution times. In the latter case, ADINA interpolates the forces to the ADINA solution times and writes the interpolated data to unit 58.

The forces read from unit 58 are added to the forces from the other specified loadings.

At the beginning of the solution analysis (or at the beginning of the restart run), the total nodal point forces are always defined by the displacement, velocity and accelerations at the beginning of the solution analysis (or at the beginning of the restart run), and the nodal point forces read from unit 58 are not used.

Displacements can be written to unit 58 in the format given above.

11.13 Remeshing options

- In some cases (e.g., rolling, forging), a finite element mesh might become distorted to such an extent that the solution stops. A restart analysis cannot be used to continue the solution; usually a new mesh has to be created in order to solve the problem. This section describes the remeshing options available in ADINA.
- The mapping option allows you to map the last obtained solution from the current mesh onto a new mesh. The mapping option is applicable to 2-D solid elements in large strain analysis only, when the plastic-bilinear or plastic-multilinear material models are used. Note that 2-D elements can only be used in the Y-Z plane.

- When you anticipate that you will use the mapping option, you must create a mapping file (extension .map) in the first run. In the AUI user interfaces, choose Control→Mapping and select “Create Mapping File” or from the AUI command line, use the command MASTER MAP-OUTPUT=YES. You must also save the AUI database file.
- To create a new mesh on the deformed geometry and continue the analysis, follow these steps:
 - ▶ Open the AUI database file.
 - ▶ In the AUI user interfaces, choose Control→Mapping and select “Remesh and Continue Analysis”, or from the AUI command line, use the command MASTER MAP-OUTPUT = REMESH.
 - ▶ Read the nodal deformations into the AUI. The nodal deformations are stored in a file with extension .mds). To read the nodal deformations in the AUI user interfaces, specify the nodal deformations file in the Control→Mapping dialog box, or from the AUI command line, use the command MASTER NODAL-DEFORMATION-FILE = (filename).
 - ▶ When the AUI reads the nodal deformations, it creates new geometry lines corresponding to the boundaries of the deformed body. Note, in this AUI version, the AUI does not create 3-D deformed surfaces.
 - ▶ Now use the dialog boxes and/or commands of the AUI to create a new mesh. Set the solution start time to the last obtained solution time. You must use the same element types, material models and contact conditions as you used in the previous mesh. However, you can modify the loading, time stepping and boundary conditions.
 - ▶ Create a new data file and run ADINA. You do not have to provide the restart file from the previous run to the current ADINA run, but you must provide the mapping file from the previous run to the current ADINA run.

11.14 Analysis zooming

- The analysis zooming feature is used to analyze details of models starting from a relatively coarse mesh and subsequently zooming in on specific regions of interest. The feature is available in the ADINA program only.
- The initial (“coarse model”) analysis requires the user to specify that the analysis zooming feature might be used in subsequent runs. All other options are the same as usual. The ADINA program creates a mapping file which can be used in subsequent layered models.
- The analysis zooming feature is based on St. Venant’s principle that if an actual distribution of forces is replaced by a statically equivalent system, the distribution of stresses and strains is altered only near the regions of load applications.
- The advantages of this zooming method are
 - ▶ the need for transition regions is eliminated
 - ▶ a user can change the zoomed region and still use the full model initial solutions for every zooming calculation, thereby enabling the user to experiment with different designs of the region of interest
 - ▶ the method can help to demonstrate the adequacy of mesh refinements.
- When using the analysis zooming feature, note that
 - ▶ A user can zoom on any part of the model. Zooming can be repeated on the already zoomed parts. For every layer, a new mapping file is created, which can be used in further zooming on the model.
 - ▶ There can be more than one zoomed part.
 - ▶ Boundary conditions, material models and loading are derived from the initial model, but they can be modified.

- ▶ The *.dat* file for the zoomed part must have a different name from the full model *.dat* file.
- ▶ Only 2-D solid and 3-D solid elements can be used in the zoom modeling.
- ▶ For 2D solid elements, the node numbers can be used for zooming analysis include 3, 4, 6, 8 and 9. For 3D solid elements, the node numbers can be used for zooming analysis include 4, 8, 10, 11, 20 and 27.
- ▶ ADINA performs a step-by-step analysis. The *.map* file from the full model solution must be used in the analysis of the zoomed model. Although this is not a restart analysis, the *.map* file should be regarded as the restart file as in a standard restart run.
- ▶ In the analysis zooming method, for every time step, the displacements from the solution of the initial model are mapped as prescribed displacements on the boundary nodes of the zoomed model. In thermo-mechanical coupling analysis the temperatures from the solution of the initial model are also mapped as prescribed temperatures on the boundary nodes of the zoomed model.
- ▶ The user should be able to verify that the distance between the cut boundaries and the stress concentration region is adequate. This can be done by using the AUI (ADINA-PLOT). The AUI can overlay stress contour plots of the zoomed model on the same stress contour plots of the whole (coarse) model. In other words, the AUI can display the zoomed model on the (coarse) full model for all plotting options, namely, band plots and vector plots.
- ▶ The cut boundaries are assumed to be far enough away from stress concentration regions.
- ▶ The ATS method can be used in analysis zooming, both for the initial model and the zoomed model. If ATS is used for the zoomed model, the program maps the scaled solution of the

initial modal to boundary nodes of the zoomed model when the time step is cut-back.

- ▶ When the ATS method is used in analysis zooming, the program forces the analysis to proceed through all user-specified solution times (both for the initial model and the zoom model). That is, the program forces the parameter `RESTORE=ORIGINAL` in the `AUTOMATIC TIME-STEPPING` command.
- ▶ Time steps in the zoomed model can be smaller or equal to the time steps used in the global model.
- ▶ Analysis zooming can be used in ADINA TMC with heat-flow only, one-way TMC, iterative TMC, and direct TMC.

11.15 Stiffness stabilization

- During the solution of equations in static analysis, zero pivots may arise, for example in the following cases:

Unsupported body: If the forces acting on the body are not in equilibrium, one or more rigid body motions of the body are activated and no solution can be expected. Even if the forces acting on the body are in equilibrium, so that no rigid body motion is in fact activated, zero pivots are present corresponding to the rigid body modes.

Contact analysis, in which one or more of the individual parts of the model (not considering contact) contain rigid body motions. When the parts are not in contact, then there is nothing to prevent the rigid body motions. (This includes tied contact.)

Mesh glueing, when one or more of the individual parts of the model (not considering glueing) contain rigid body motions.

General constraints, when one or more of the individual parts of the model (not considering the general constraints) contain rigid body motions.

These zero pivots will stop the solution, unless the zero pivots are prevented from occurring.

- Stiffness stabilization is used to prevent the equation solver from encountering zero pivots. Stiffness stabilization is available for static analysis, with or without low-speed dynamics. (In dynamic analysis, the zero pivots are not present due to the mass matrix, so stiffness stabilization is not available for dynamic analysis.)
- Parts of the model with rigid body motions can alternatively be treated by adding weak springs at various locations in the model. The advantages of using stiffness stabilization, instead of using weak springs, are:
 - ▶ Determining the number, location and stiffness of the springs requires a lot of user intervention.
 - ▶ There may be no suitable locations for the springs.
 - ▶ The stiffness of each spring has to be entered as an absolute value (with dimensions of force/length) while the stiffness stabilization factor is dimensionless (see below).
 - ▶ The springs generate internal forces which affect the final solution while stiffness stabilization does not affect the internal forces. It is sometimes hard to assess how much the springs affect the final solution.
- Stiffness stabilization modifies the diagonal stiffness terms (except for those belonging to contact equations) as follows:

$$\mathbf{K}_{ii} = (1 + \varepsilon_{STAB}) \mathbf{K}_{ii}$$

where ε_{STAB} is a dimensionless stabilization factor. The right-hand side load vector is not modified. If a diagonal stiffness term is zero before factorization, the term remains zero.

- There are four stiffness stabilization options available, which are selected using the STABILIZE and STABFACTOR parameters of the MASTER command:

STABILIZE=NO (no stabilization)
STABILIZE=YES (stabilization, with ε_{STAB} = STABFACTOR)
STABILIZE=AUTOMATIC (stabilization is activated if needed)
STABILIZE=NONLINEAR (stabilization for linear analysis only)

The defaults are STABILIZE=NO, STABFACTOR=1E-10.

- When STABILIZE=AUTOMATIC, the use of stabilization is determined based on the ratio of the factorized maximum and minimum diagonals of the stiffness matrix. This determination is made for every equilibrium iteration in nonlinear analysis. When stabilization is used, stabilization is applied to all degrees of freedom using the value ε_{STAB} (as if STABILIZE=YES).

STABILIZE=AUTOMATIC cannot be used in the DMP version.

- When STABILIZE=NONLINEAR, the program uses stabilization for nonlinear analysis only.
- In linear analysis, stabilization should be used with caution, since the right-hand-side load vector is not modified. The solution can therefore be affected by stiffness stabilization.

It is recommended to try the analysis first without stabilization. If the equation solver encounters zero pivots, then try one of the following methods:

- Use stabilization with the smallest possible value of ε_{STAB} for which the equation solver gives a solution, or
 - Use stabilization, and change the analysis to a nonlinear analysis, for example specify element birth-death in one of the element groups; or specify a nonlinear material, with material constants chosen so that the material response is linear (for example, an elastic-plastic material with a very high yield stress).
- In nonlinear analysis, since the right-hand-side load vector is not modified, the final converged solution is the same as without

stabilization (assuming that the tolerances are tight enough). However, the rate of convergence can be worsened due to the stiffness stabilization, so that more equilibrium iterations are required.

- Stiffness stabilization is only useful for the sparse and 3D-iterative solvers. The iterative multigrid solver does not fully factorize the stiffness matrix and hence cannot properly trigger the automatic stabilization.
- Stiffness stabilization is not available in FSI analysis with direct coupling.

11.16 Geometric imperfections

- ADINA supports adding geometric imperfections to a structure prior to analysis. The initial coordinates of the nodes are modified based on the imperfection, and no strains (nor stresses) are generated.
- The imperfections can be obtained from a mode shape generated from a linearized buckling analysis, or a frequency analysis, or from nodal displacements generated from a static or dynamic analysis.
- Initial geometric imperfections can be useful for transient dynamic post-buckling type problems where geometric perturbations help trigger the buckling modes. The initial geometric imperfection in such cases should ideally be generated from a linearized buckling analysis.
- Initial geometric imperfections can also be useful in obtaining an approximate stress-free configuration for flexible structures such as tents. In this case, a static analysis can be performed with gravity loading, and the displacements of that analysis used as initial geometric imperfections for subsequent analyses.

11.17 Bolt feature

Overview

- The bolt option is a modeling feature which uses element groups with specific bolt-type loadings to simulate different bolt handling procedures. The supported bolt-type loadings are:

1. bolt tensioning, i.e., the axial force in the bolt is specified.
2. bolt shrinkage, i.e., the bolt shortening (reduction in bolt length) is specified.

In both of these cases, the deformations of the rest of the structure are considered. For example, if a single bolt is loaded with a user-specified bolt tension, then, after the bolt loading is applied, the axial force of the bolt has the given value, and the axial force acts upon the rest of the structure, causing structural deformations.

- Different modeling techniques with varying complexity can be used to model a bolt. Three such techniques are shown in Fig. 11.17-1, in increasing level of complexity.
- Any number of bolts can be included in the model. The bolts can all be loaded simultaneously, or the bolts can be loaded sequentially.
- Bolt type loadings are applied during special "bolt loading" iterations. During the bolt loading iterations, time is frozen and the bolt parameters are automatically iteratively adjusted as described below. After the bolt loading iterations have converged, then the time steps proceed as usual.

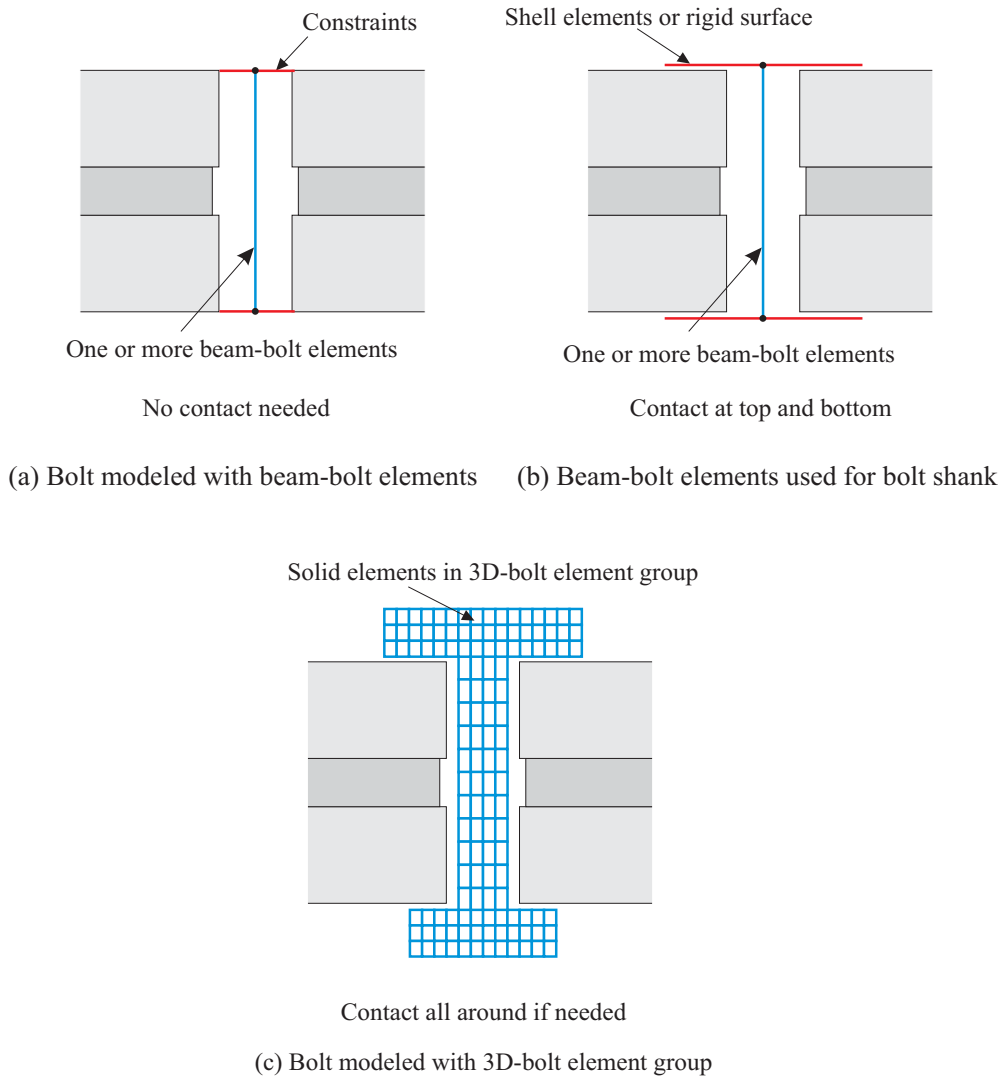


Figure 11.17-1: Different bolt modeling techniques

The bolt loading can be performed:

- at the beginning of the analysis, i.e. at time zero.
- at the beginning of a restart analysis
- at any given solution time.

- A bolt iteration consists of the same process as is normally used for static analysis, namely the solution of the equilibrium equations as discussed in Sections 7.1 and 7.2. In each iteration all elements, including bolt elements, contribute to the global force vector, and the increment in displacements, including the displacements of the bolt element nodes, is obtained.

The remainder of the structure (that is all of the rest of the model, including bolts not being loaded) behaves exactly as usual. For example, plasticity can occur. Therefore, it is possible that the bolt iterations might not converge. In this case, the bolt loading can be divided into sub-steps, so that the change in bolt loads is smaller during each sub-step.

11.17.1 Beam-bolt element

- The beam-bolt element is implemented as an elastic Hermitian beam element with a modified axial force - length relationship. The formulation used for the beam-bolt elements can either be small displacements or large displacements. Any cross-section available for the beam element can be used. However, the material model must be linear elastic.
- The axial force - length relationship of the beam-bolt element is

$$F = k(^tL - L_b) + F_b \quad (11.17-1)$$

where F is the bolt axial force, k is the bolt axial stiffness EA/L_b , tL is the bolt current length, L_b is the bolt adjusted length (defined below) and F_b is the bolt force corresponding to L_b . (Thermal effects are ignored in this discussion.) The bolt adjusted length is defined as $L_b = {}^0L - \Delta_b$, where 0L is the bolt original length and Δ_b is the bolt shortening.

Using $^tL = {}^0L + {}^t\Delta$ where 0L is the bolt initial length and ${}^t\Delta$ is the bolt axial displacement gives

$$F = k(^t\Delta + \Delta_b) + F_b \quad (11.17-2)$$

Initially (before the first bolt-loading is applied), F_b and Δ_b are zero, which means that $F = k^t \Delta$ and the element behaves exactly like a usual beam element.

- We now discuss the details of the bolt iterations for beam-bolt elements. For simplicity, in this discussion, we show only a single bolt element, and we assume that the bolt load is applied at time 0, that is, before any other deformations have occurred.
- The process of bolt tensioning is shown in Figure 11.17-2. During the bolt iterations, the axial stiffness of the bolt element is set very small (to $f_1 k$, where $f_1 \ll 1$), and the axial force in the bolt element is replaced by the user-specified axial force F_b . The bolt shortening Δ_b is determined from the condition $\Delta_b = -^t \Delta$ in which $^t \Delta$ is determined from the displacements of the bolt element nodes. The bolt iterations continue until equilibrium is satisfied and also the condition

$$\frac{\left| \Delta_b^{(ite)} - \Delta_b^{(ite-1)} \right|}{\max \left(\left| \Delta_b^{(ite-1)} \right|, \left| \Delta_b^{(ite)} \right| \right)} < \text{tol}$$

is satisfied. At this point, the bolt axial stiffness is restored to its normal value. Since $^t \Delta + \Delta_b = 0$, the bolt axial force F computed from (11.17-2) equals the user-specified axial force F_b .

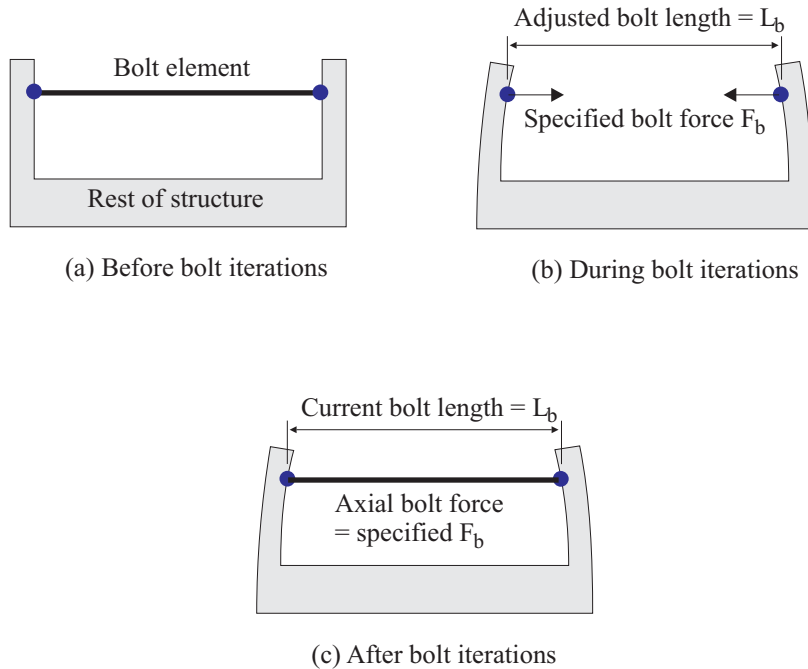


Figure 11.17-2: Bolt force loading

• The process of bolt shrinkage is shown in Figure 11.17-3. During the bolt iterations, the axial stiffness of the bolt element is set very large (to $f_2 k$, where $f_2 \gg 1$) and the axial length of the bolt element is set to $L_b = {}^0L - \Delta_b$, where Δ_b is user-specified. Since the bolt still has some flexibility, the bolt contributes axial force to the nodes using $F = f_2 k({}^t\Delta + \Delta_b)$, where, since the axial stiffness is very large, ${}^t\Delta + \Delta_b \approx 0$. The bolt iterations continue until equilibrium is satisfied and also the condition

$$\frac{|F_b^{(ite)} - F_b^{(ite-1)}|}{\max(|F_b^{(ite-1)}|, |F_b^{(ite)}|)} < \text{tol}$$

is satisfied. At this point, the bolt axial stiffness is restored to its usual value and the bolt axial force F_b is set to F . Again, for

${}^t\Delta + \Delta_b = 0$, the bolt axial force F computed from (11.17-1) equals the axial force F_b .

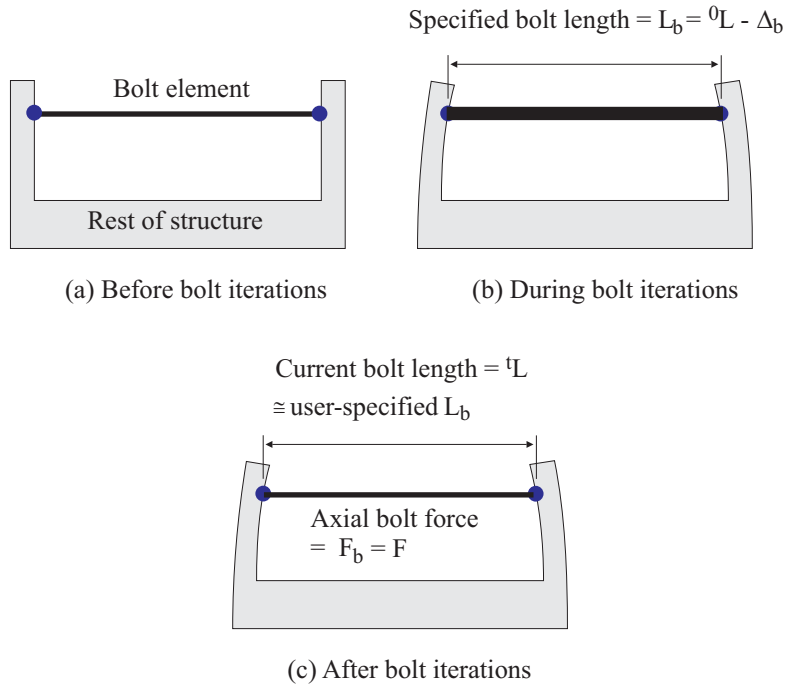


Figure 11.17-3: Bolt shrinkage loading

- For all other iterations, in which the bolt is not being adjusted, the bolt axial force - length relationship is given by (11.17-1), where F_b and Δ_b are in general non-zero and are held constant.
- In the beam-bolt element, no additional global equations are used to include the additional bolt variables Δ_b and F_b .
- The bolt tolerance in the above equations is set by the BOLT-OPTIONS TOLERANCES parameter, or by the EGROUP BEAM BOLT-TOL parameter. The default value is 0.01.

- The values of f_1 and f_2 in the above equations is given by BOLT-OPTIONS FACTOR1 and BOLT-OPTIONS FACTOR2 respectively. The default values are FACTOR1=1E-6, FACTOR2=1E6.

11.17.2 3D-bolt element group

- An entire 3D element group can be designated as a 3D-bolt element group. The elements in this group can use any formulation (small or large displacements, small or large strains), any material model and any number of nodes per element. The mixed u/p formulation or the incompatible modes formulation can also be used.

The geometry of the 3D-bolt element group is also arbitrary. The element sides and faces can be straight or curved.

- Each bolt is modeled as a separate 3D-bolt element group. Multiple bolts cannot be modeled within a single 3D-bolt group.
- The basic ideas used in the 3D-bolt are shown in Fig 11.17-4. This figure shows a very simple bolt model of four elements, with the ends of the bolt fixed. For simplicity, it is assumed that the Poisson's ratio is zero, so that the cross-section of the bolt doesn't change during the deformations.

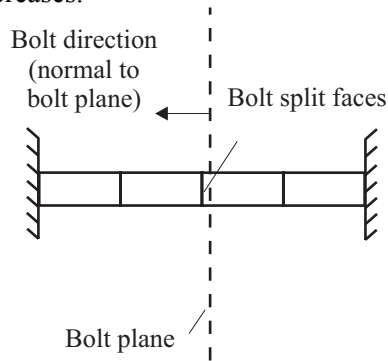
Fig. 11.17-4(a) shows the 3D-bolt group along with its bolt plane. This bolt plane is an additional input for the 3D-bolt group. As seen, the bolt plane determines the bolt direction and the bolt split faces. (For more information about the bolt plane, see Section 11.17.5.)

Fig 11.17-4(b) shows an incompatible displacement, called the bolt displacement, applied to the nodes on the bolt split faces. Because the bolt model is fixed at its ends, the elements must increase in length, and, to satisfy equilibrium, the elements must all be under uniform tension. The nodal point forces acting at the nodes on the bolt split faces are combined into a single bolt force.

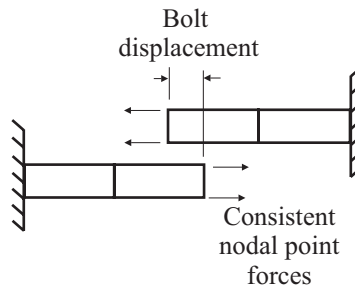
Fig 11.17-4(c) shows the bolt element group after deformations, as visualized during postprocessing. The elements appear to be different lengths because the bolt displacement is not included in the plot. However, the strains are the same in all elements.

In this example, it is clear that the bolt force is determined by

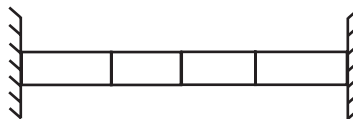
the bolt displacement, and that the bolt force increases as the bolt displacement increases.



(a) 3D-bolt group with bolt plane



(b) Bolt displacement and consistent nodal point forces



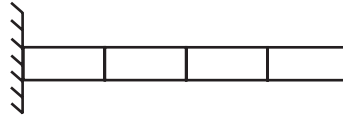
(c) Bolt element group, as visualized

Figure 11.17-4: 3D-bolt element group example

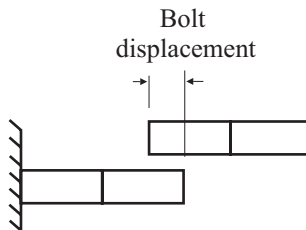
- Fig 11.17-5 shows the same problem, but with one end of the bolt free. Now when the bolt displacement is applied, the elements

on the free end move stress-free, and the resulting plot is shown in Fig 11.17-5(c). This example justifies the use of the term "bolt displacement".

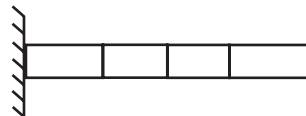
When the bolt end is free, it is clear that the bolt force is zero for any value of the bolt displacement.



(a) Before deformations



(b) After deformations



(c) As visualized

Figure 11.17-5: 3D-bolt element group example, free end

- In the general case, a 3D-bolt group has two additional quantities: the bolt displacement (denoted v_b) and the bolt force (denoted F_b). These quantities are used in the global solution of equations as follows:

1) Bolt force specified:

$$\begin{bmatrix} \mathbf{K} & \mathbf{KUB} \\ \mathbf{KBU} & \mathbf{KBB} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{u} \\ \Delta v_b \end{bmatrix} = \begin{bmatrix} \mathbf{R} - \mathbf{F} \\ R_b - F_b \end{bmatrix} \quad (11.17-3)$$

where \mathbf{K} is the usual global stiffness matrix, \mathbf{R} is the usual external loads vector, \mathbf{F} is the usual internal loads vector, $\Delta \mathbf{u}$ is the usual increment in displacements, \mathbf{KBU} , \mathbf{KUB} , \mathbf{KBB} are coupling matrices, and R_b is the specified bolt force. During the equilibrium iterations, both $\Delta \mathbf{u}$ and Δv_b are obtained, and at equilibrium, $F_b = R_b$. Thus the value of v_b that results in the specified R_b is obtained by this procedure.

In addition to the usual convergence criteria, the convergence criterion

$$\frac{|R_b - F_b|}{\max(|R_b|, |F_b|)} < \text{tol}$$

is used.

2) Bolt displacement specified:

$$\begin{bmatrix} \mathbf{K} & \mathbf{KUB} \\ \mathbf{KBU} & \alpha \mathbf{KBB} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{u} \\ \Delta v_b \end{bmatrix} = \begin{bmatrix} \mathbf{R} - \mathbf{F} \\ \alpha (v_b^{presc} - v_b) \end{bmatrix} \quad (11.17-4)$$

where v_b^{presc} is the specified bolt displacement and α is very large.

For convergence, v_b must be equal to v_b^{presc} . This procedure is analogous to the procedure in which displacements are prescribed.

3) Usual equilibrium iterations in which the bolt is not being adjusted:

$$\begin{bmatrix} \mathbf{K} & \mathbf{KUB} \\ \mathbf{KBU} & \alpha \mathbf{KBB} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{u} \\ \Delta v_b \end{bmatrix} = \begin{bmatrix} \mathbf{R} - \mathbf{F} \\ 0 \end{bmatrix} \quad (11.17-5)$$

The last row of this system of equation results in $\Delta v_b = 0$, so the global equilibrium equation reduces to the usual case.

In addition to the usual convergence criteria, the convergence criterion

$$\frac{|v_b^{presc} - v_b|}{\max(|v_b^{presc}|, |v_b|)} < \text{tol}$$

is used.

- The bolt tolerance in the above equations is set by the BOLT-OPTIONS TOLERANCES parameter, or by the EGROUP THREEDSOLID BOLT-TOL parameter. The default value is 0.01.
- When there are several 3D-bolts, then each bolt contributes an additional equation to the global system of equations. There is no limit to the number of 3D-bolts that can be present in the model.

11.17.3 Usage of bolt loadings

- The BOLT-OPTIONS command is used to control general bolt options, including the use of bolt tables. There are two basic options:

BOLT-OPTIONS TABLE=NO

Then the type of bolt loading is specified by BOLT-OPTIONS TYPE (either FORCE or LENGTH) and bolt loading occurs at solution time BOLT-OPTIONS TIME. The number of bolt loading steps is specified by BOLT-OPTIONS STEPS. All bolts in the model are loaded simultaneously.

For example

```
BOLT-OPTIONS TABLE=NO TYPE=FORCE TIME=2.0  
STEPS=2
```

At time 2.0, all bolts are loaded using force loading. In the first bolt-loading step, half of the bolt load is applied, and the second bolt-loading step, the entire bolt loading is applied.

BOLT-OPTIONS TABLE=YES

Then the BOLT-TABLE commands are used to specify the type of bolt loading, the solution time for the bolt loading, and also the sequence in which the bolts are loaded. For example

```
BOLT-OPTIONS TABLE=YES  
BOLT-TABLE 1 TYPE=FORCE TIME=2.0  
1 2 0.5  
1 3 0.2  
2 4 0.3  
3 2 1.0  
3 3 1.0  
3 4 1.0
```

At time 2.0, bolts 2, 3 and 4 are loaded using force loading. In the first bolt-loading step, bolt 2 is loaded by 0.5 times its specified bolt force, bolt 3 is loaded by 0.2 times its specified bolt force and bolt 4 is not adjusted. In the second bolt loading step, bolts 2 and 3 are not adjusted, and bolt 4 is loaded by 0.3 times its specified bolt force. In the third bolt loading step, all three bolts are loaded by their specified bolt forces.

- The results at any bolt step can be saved to file. If bolt tables are used, the saved bolt steps all have the same solution time, and are distinguished by their bolt sequence numbers.
- Bolt loading can change from bolt tensioning to bolt shrinkage at different stages of the analysis. This is achieved using bolt tables with different bolt loading types.
- Bolt loading can be performed in static or implicit dynamic analysis.

- Bolt iterations that are performed before the first time step are always static or low-speed dynamic, never truly dynamic, so as to not introduce any dynamic effects during bolt tightening such as ringing. Note that in practice bolt tightening is typically a quasi-static process.
- The special treatment of bolt iterations applies only to the bolt iterations that are performed before the first time step. Bolt iterations that are performed after the first time step are treated exactly the same as other time steps. For example, in dynamic analysis, bolt iterations performed before the first time step are static or low-speed dynamic, but bolt iterations performed after the first time step are dynamic.
- In a restart analysis, the bolt iterations are treated exactly the same as other time steps, since there were already time steps performed prior to the start of the restart analysis.
- Damping can be applied to the model during the bolt iterations before the first time step using the BOLT-OPTIONS DAMPING parameter. This can be useful to stabilize any rigid-body motions that may be present before contact is established. It is possible to apply damping only during the bolt iterations, not to the time stepping, such that the solution at the end of the time step is free of any damping.
- In static analysis, a static analysis is performed during the bolt iterations if bolt damping = 0. If bolt damping > 0, a low-speed dynamic analysis is performed during the bolt iterations before the first time step with the specified bolt damping factor and an inertia factor = 1, and with the Newmark time integration method. After the first time step, a static analysis is performed during the bolt iterations, irrespective of the bolt damping factor.
- In low-speed dynamic analysis, a low-speed dynamic analysis is performed during the bolt iterations with the same settings as during the time steps if bolt damping = 0. If bolt damping > 0, the bolt damping factor overrides the low-speed dynamics damping factor during the bolt iterations before the first time step. After the first time step, a low-speed dynamic analysis is performed during the bolt iterations with the same settings as during the time steps,

irrespective of the bolt damping factor.

- In dynamic analysis, a static analysis is performed during the bolt iterations before the first time step if bolt damping = 0. If bolt damping > 0, a low-speed dynamic analysis is performed during the bolt iterations before the first time step with the specified bolt damping factor and an inertia factor = 1, and with the same time integration method as for the dynamic analysis during the time steps. After the first time step, a dynamic analysis is performed during the bolt iterations with the same settings as during the time steps, irrespective of the bolt damping factor. Hence, in dynamic analysis, bolt damping is often required to remove rigid body modes during the bolt iterations before the first time step.
- Table 11.17-1 shows the behavior of bolt damping.

Table 11.17-1: Behaviour of bolt damping

	Bolt damping = 0	Bolt damping > 0
True dynamics: MASTER ... ANALYSIS=DYNAMICS, no ATS	Before First Time Step: Bolt iterations use static analysis without low-speed dynamics and without ATS.	Before First Time Step: Bolt iterations use low-speed dynamics with the user-specified time integration method and the specified bolt damping factor, with ATS forced on.
	After First Time Step: Bolt iterations use true dynamics analysis.	After First Time Step: Bolt iterations use true dynamics analysis, no bolt damping is applied, no ATS.
True dynamics: MASTER ANALYSIS=DYNAMICS, ATS	Before First Time Step: Bolt iterations use static analysis without low-speed dynamics, but with ATS.	Before First Time Step: Bolt iterations use low-speed dynamics with the user-specified time integration method and the specified bolt damping factor, with ATS.
	After First Time Step: Bolt iterations use true dynamics analysis with ATS.	After First Time Step: Bolt iterations use true dynamics analysis, no bolt damping is applied, with ATS.

	Bolt damping = 0	Bolt damping > 0
Statics, no ATS, no low-speed dynamics	Bolt iterations use static analysis without low-speed dynamics and without ATS.	Before First Time Step: Bolt iterations use low-speed dynamics with the Newmark method and the specified bolt damping factor, with ATS forced on.
		After First Time Step: Bolt iterations use static analysis without low-speed dynamics, no bolt damping is applied, and without ATS.
Statics, ATS, no low-speed dynamics	Bolt iterations use static analysis without low-speed dynamics and with ATS.	Before First Time Step: Bolt iterations use low-speed dynamics with the Newmark method and the specified bolt damping factor, with ATS.
		After First Time Step: Bolt iterations use static analysis without low-speed dynamics, no bolt damping is applied, but with ATS.
Statics, low-speed dynamics (ATS is always on for low-speed dynamics).	Bolt iterations use low-speed dynamics with the same settings as during the time steps, and with ATS.	Before First Time Step: Bolt iterations use low-speed dynamics with the same time integration method as during the time steps, but with the specified bolt damping factor, with ATS.
		After First Time Step: Bolt iterations use low-speed dynamics with the same settings as during the time steps, and with ATS.

Explicit dynamic analysis can be performed after a restart from bolt loading implicit analysis.

11.17.4 Usage of beam-bolts

- A beam element group is identified as a bolt element group using the parameter `OPTION=BOLT` in the `EGROUP BEAM` command. The bolt element group is assigned a bolt number using the `BOLT-NUMBER` parameter. The bolt load (either force or shortening, depending upon `BOLT-OPTIONS` and `BOLT-TABLE`) is specified using the `BOLT-LOAD` parameter.

A beam-bolt element group can contain one or more elements. If the group contains more than one element, it is assumed that the

elements are all connected sequentially and all have the same length, as shown in Figure 11-17.6.

It is possible to have several bolts with the same bolt label.

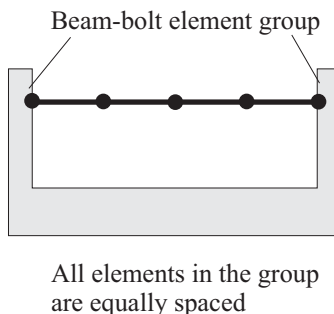


Figure 11.17-6: Modeling of a bolt using several beam-bolt elements.

11.17.5 Usage of 3D-bolts

- A 3D element group is identified as a bolt element group using the parameter `OPTION=BOLT` in the `EGROUP THREEDSOLID` command. The bolt element group is assigned a bolt number using the `BOLT-NUMBER` parameter. The bolt load (either force or shortening, depending upon `BOLT-OPTIONS` and `BOLT-TABLE`) is specified using the `BOLT-LOAD` parameter.

- The 3D bolt element group also uses a bolt plane as part of its definition. In the general case, the bolt plane is defined using the `BOLT-PLANE` command and the bolt plane is specified using the `BOLT-PLANE` parameter of the `EGROUP THREEDSOLID` command.

The bolt plane gives the bolt direction and the approximate location of the bolt split faces. The bolt direction is used to determine the direction of the bolt displacement and the direction of the bolt force. The bolt plane is used to determine the bolt split faces. Note that the bolt split faces do not necessarily coincide with the bolt plane. Rather, the program determines the bolt split faces to lie "near" the bolt plane.

- It is necessary for the bolt plane to intersect the elements, in

such a way that the bolt direction is aligned with the bolt, and the entire bolt force is transmitted across the bolt plane. Fig 11.17-7 shows some examples of incorrect and correct definitions.

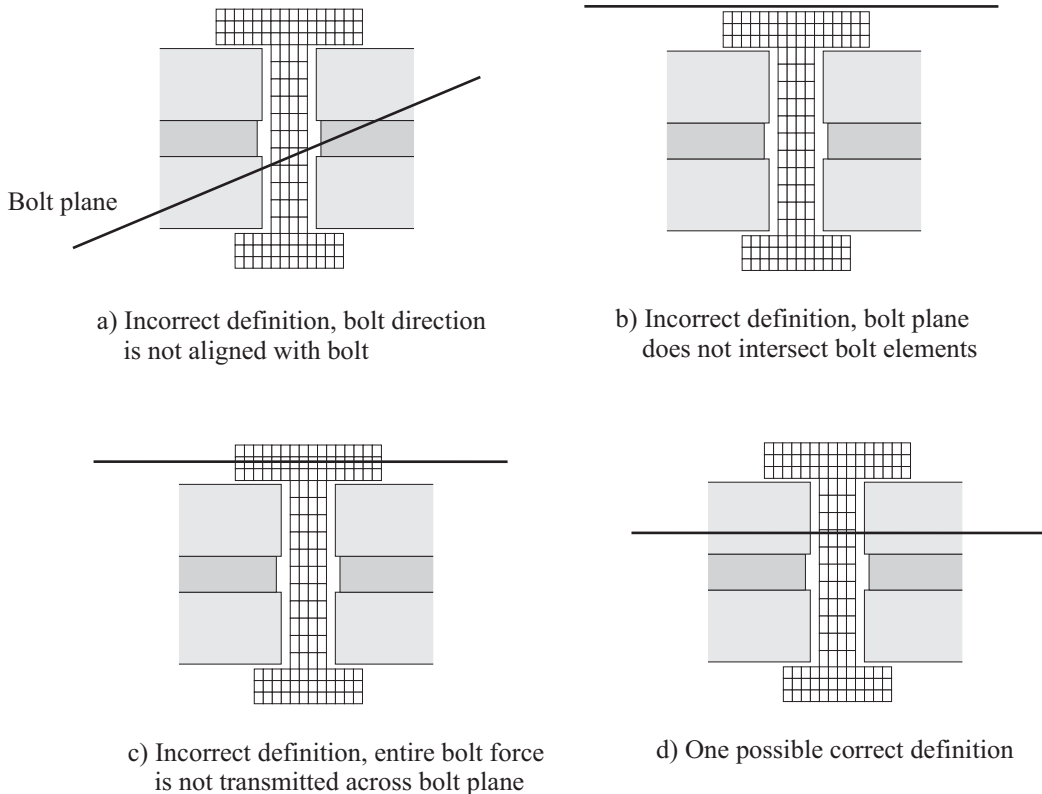


Figure 11.17-7: Examples of bolt plane definition

- The following options are available for the BOLT-PLANE command, see Fig 11.17-8:
 - OPTION=TWOPOINT: The bolt-plane is specified by two geometry points and a factor giving the placement of the bolt plane relative to the two points. Factor=0 corresponds to the bolt-plane passing through the first point and factor=1 corresponds to the bolt-plane passing through the second point.
 - OPTION=TWONODE: Similar to TWOPOINT, except that two nodes are used.
 - OPTION=XPLANE: The bolt plane has normal in the x

direction and is positioned at the given X coordinate.

OPTION=YPLANE or ZPLANE: Similar to XPLANE, but using the y and z coordinate axes instead.

OPTION=POINTAUTO: The bolt plane is positioned at the given geometry point, and the bolt plane direction is automatically determined.

OPTION=NODEAUTO: Similar to POINTAUTO, except that a node is used.

OPTION=POINTNORMAL: The bolt plane is positioned at the given geometry point, and the bolt plane direction is specified using the given normal.

OPTION=NODENORMAL: Similar to POINTNORMAL, except that a node is used.

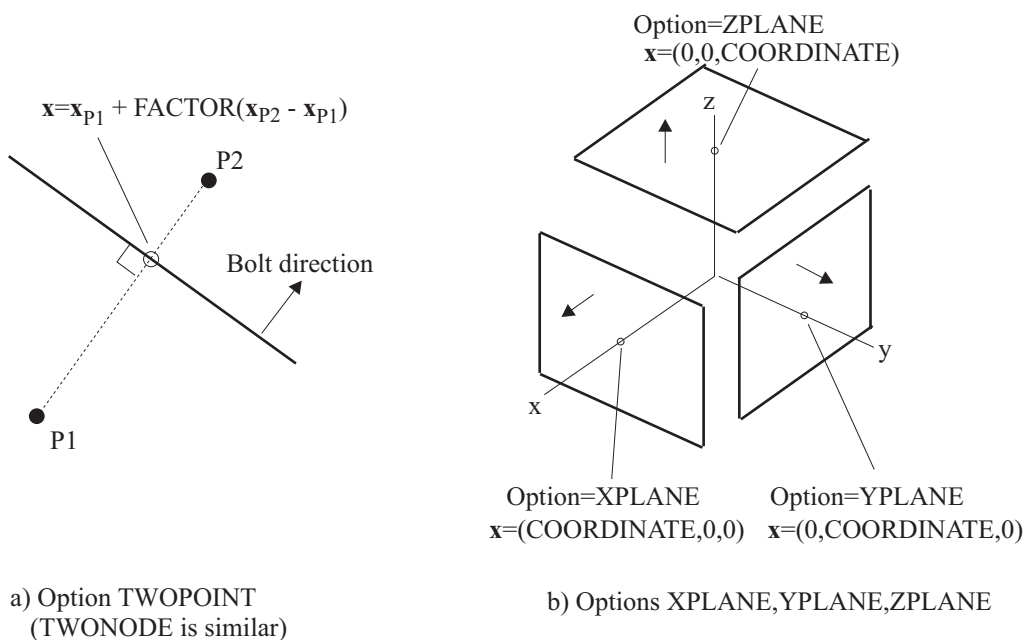


Figure 11.17-8: Options for bolt plane definition

- It is also allowed to define a 3D-bolt group without using a bolt plane. This is done by setting BOLT-PLANE=0 in the EGROUP THREEDSOLID command.

In this case, the AUI automatically determines the bolt plane as follows (Fig 11.17-9). First the centroid and moments of inertia of the bolt group are obtained (assuming unit density). Then the bolt

direction is obtained as the direction of the minimum principal moment of inertia, and the bolt plane passes through the centroid. These calculations are all done during data file generation.

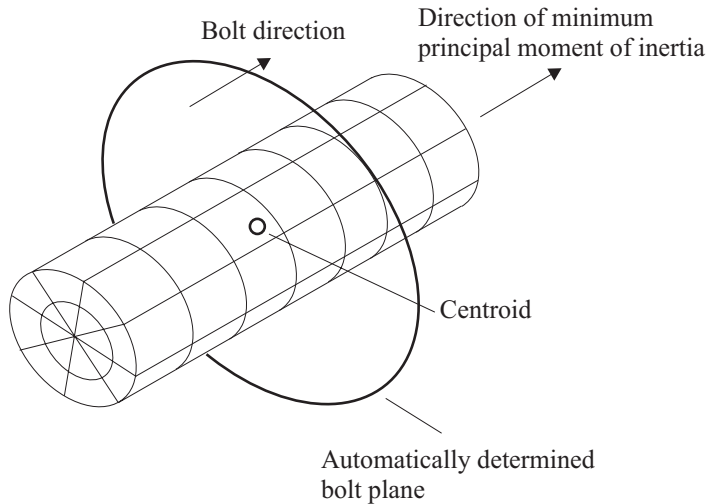


Figure 11.17-9: Automatic determination of bolt-plane

The inertial properties are determined using numerical integration within the finite element mesh. Because the mesh itself is used, the inertial properties might not exactly match the properties corresponding to the underlying geometry. For this reason, if the direction of minimum principal moment of inertia is within 1 degree of a global coordinate direction, the global coordinate direction is used instead.

This algorithm gives a reasonable choice for the bolt plane in many cases.

- The AUI outputs the bolt direction and bolt cross-sectional area for each bolt during data file generation. This information can be used to confirm that the bolt plane is defined correctly.
- When the bolt force is specified for a system in which one end of the bolt is free (for example, the system shown in Fig 11.17-5), the global system of equations (11.17-3) is singular. This is physically correct as there is no non-zero bolt force that can satisfy equilibrium.

To prevent numerical difficulties in this case, parameter BOLT-

OPTIONS FACTOR3 can be used. FACTOR3 provides an additional stiffness relating the bolt force and bolt displacement displacement. The default for FACTOR3 is zero, and the dimension of FACTOR3 is stiffness. If FACTOR3 is set nonzero, its value should be much smaller than the bolt stiffness itself.

Alternatively, stiffness stabilization can be used to prevent numerical difficulties.

- During large displacement analysis, the bolt direction may change during the solution process. The program computes the updated bolt direction using three nodes on the bolt split faces; as these three nodes move, the bolt direction is correspondingly updated.
- One pitfall when importing Nastran files for the model definition is the following. Elements with different numbers of nodes are placed into different element groups when importing Nastran files using the option SPLIT=YES. But the 3D-bolt group feature assumes that all elements modeling the bolt are placed in the same element group. Therefore, when importing Nastran files, the option

NASTRAN-ADINA ... SPLIT=NO

(which is the default) should be used, when elements with different numbers of nodes are used to model the bolt.

- The AUI determines the bolt split faces during generation of the ADINA dat file. Any errors or warnings concerning the bolt split faces are output then.
- It is recommended that the meshing of the 3D-bolt group be compatible, in the sense that all of the adjacent internal element faces must match.

The reason for this recommendation is that the compatibility of the mesh is assumed when determining the bolt split faces. If the mesh is incompatible, the bolt split faces might not be determined correctly.

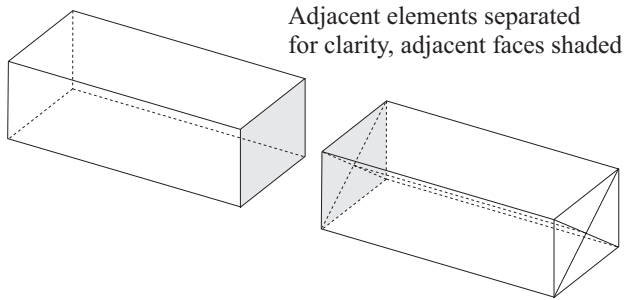
Three different cases are shown in Fig 11.17-10. In the first case (Fig 11.17-10a), the mesh is incompatible, and there is an unmatched node on one of the adjacent element faces. If this mesh

was used in a 3D-bolt group, the AUI would not be able to determine the bolt split faces.

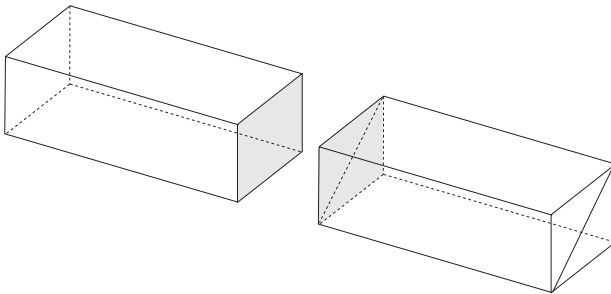
In the second case (Fig. 11.17-10b), the mesh is, strictly speaking, incompatible; however the AUI recognizes that two triangular faces can match an adjacent quadrilateral face, so the AUI considers the mesh to be compatible. The second case occurs often in practice, for example, the command `GBODY NODES=8 PYRAMIDS=NO` creates only tetrahedra and hexahedra, with the triangular faces from the tetrahedra are directly adjacent to the quadrilateral faces from the hexahedra.

In the third case (Fig 11.17-10c), the mesh is totally compatible.

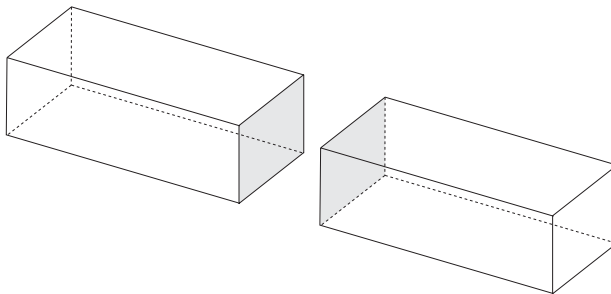
- It is possible that meshes in which some faces are compatible and other faces are incompatible might cause the AUI to determine the bolt split faces incorrectly. However the AUI checks the bolt split faces for correctness: any node on a bolt split face must be attached to elements on both sides of the bolt plane, and any node attached to elements on both sides of the bolt plane must be on a bolt split face. The AUI gives an error message if these checks are not passed.



- (a): Incompatible meshing, internal element faces do not match, some nodes on internal element faces are not matched, mesh should not be used



- (b): Incompatible meshing, internal element faces do not match, but each quadrilateral face is matched by two triangular faces, mesh is acceptable



- (c): Compatible meshing, internal element faces match, mesh is acceptable

Figure 11.17-10: Incompatible and compatible meshing for 3D-bolt groups

11.17.6 Modeling issues

- One modeling pitfall is illustrated in Fig. 11.17-11. The figure shows a bolt with contact conditions. It is intended that, after the bolt force is specified, the bolt be in contact with the rigid target. However, this model will not work as intended, for the following reason. During the bolt force iterations, the bolt provides no stiffness between the top and bottom of the bolt (points A and B in the figure). And since the bolt is not initially in contact with the target, the contact also does not provide stiffness. Therefore, in the first equilibrium iteration, point A moves downwards as a rigid body under the prescribed bolt force, and this motion is very large since there is no stiffness. The remaining equilibrium iterations will probably not converge.

To prevent this effect from occurring, make sure that the contact is established before beginning bolt force iterations. For example, set up the model so that there is a very small overlap between the contacting parts. Or specify the bolt shrinkage to establish contact, then restart and specify the bolt force.

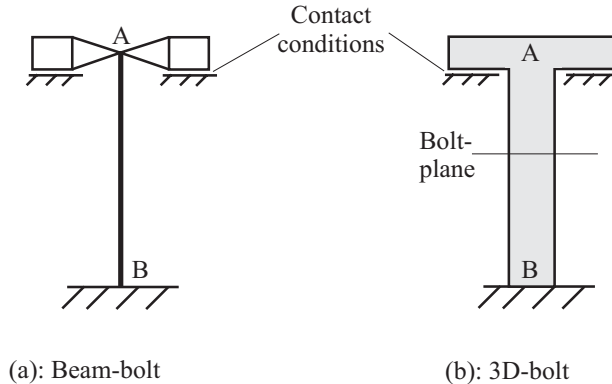


Figure 11.17-11: Bolts and contact example

11.17.7 Post-processing variables

- For each 3D-bolt, the bolt force and bolt displacement is available for post-processing using the variables BOLT-FORCE, BOLT-DISPLACEMENT. These variables can be accessed at bolt points. For example

```
BOLTPoint B2 BOLT=2  
POINTLIST B2 VAR=BOLT-FORCE BOLT-DISPLACEMENT
```

lists the bolt force and bolt displacement at bolt 2.

- For the beam-bolt elements, the bolt forces are available for post-processing using the variables BOLT_FORCE, BOLT_SHEARFORCE-S, BOLT_SHEARFORCE-T, BOLT_TORSION, BOLT_MOMENT-S, BOLT_MOMENT-T. The bolt moments are calculated at bolt local node 2. Therefore the bolt moments might not be the maximum moments in the bolt.

The bolt shortening is also available using variable BOLT_SHORTENING. These variables are stored along with the elements. For example

```
ELPOINT G5E2 GROUP=5 ELEMENT=2 OPTION=LABEL,  
        LABEL=111  
POINTLIST G5E2 VAR=BOLT_FORCE BOLT_SHORTENING
```

11.18 Component mode synthesis (CMS) method

- Component mode synthesis (CMS) is a dynamic model reduction, or domain decomposition technique used to calculate the response of large finite element systems as, for example, in the aerospace, automotive, and defense industries. The representation of the complete structure is based on using large substructures. These individual structures, or components, are typically first analyzed in detail for their frequencies and mode shapes, sometimes by different analysis groups. Then the complete structure is considered as an assemblage of components, in a CMS substructuring analysis, which can lead to a very large finite element system. Also, the reduction system matrices computed using the CMS method can be used to couple 1D multibody system programs and finite element analysis programs. An example is the use of AVL OUT and ADINA.

*ref. KJB
Section 10.3.7*

- The Craig-Bampton reduction method is used in ADINA for component mode synthesis (CMS) calculation.

ref. R.R. Craig and M.C.C. Bampton, “Coupling of

Substructures for Dynamic Analysis”, *AIAA Journal*, Vol. 6, No. 7, July 1968.

Theoretical overview

- The degrees of freedom (DOFs) of the original analysis model are divided into internal DOFs and boundary DOFs. The internal DOFs are represented by the modal displacements of interior nodes and the boundary DOFs are represented by the physical displacements of boundary nodes.
- We seek the smallest p eigenvalues and corresponding eigenvectors of the original analysis model. Let n = number of DOFs of the original analysis model where $p \ll n$, k = number of boundary DOFs (static constraint modes), q = number of fixed interface dynamic vibration modes.
- The following equations are used:

$$\begin{bmatrix} \mathbf{M}_{bb} & \mathbf{M}_{bi} \\ \mathbf{M}_{ib} & \mathbf{M}_{ii} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{u}}_b \\ \ddot{\mathbf{u}}_i \end{bmatrix} + \begin{bmatrix} \mathbf{K}_{bb} & \mathbf{K}_{bi} \\ \mathbf{K}_{ib} & \mathbf{K}_{ii} \end{bmatrix} \begin{bmatrix} \mathbf{u}_b \\ \mathbf{u}_i \end{bmatrix} = \begin{bmatrix} \mathbf{R}_b \\ \mathbf{R}_i \end{bmatrix}$$

$$\begin{bmatrix} \mathbf{u}_b \\ \mathbf{u}_i \end{bmatrix} \simeq \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{\Phi}_c & \mathbf{\Phi}_m \end{bmatrix} \begin{bmatrix} \mathbf{u}_b \\ \mathbf{q} \end{bmatrix}$$

where

\mathbf{u}_b = Physical displacements of boundary DOFs (vector of length k)

\mathbf{u}_i = Physical displacements of interior DOFs (vector of length $n - k$)

\mathbf{q} = Modal displacements of interior DOFs (vector of length \tilde{q})

$\mathbf{\Phi}_c = -\mathbf{K}_{ii}^{-1}\mathbf{K}_{ib}$ = Static constraint modes matrix (interior displacements due to unit boundary displacements, applied one at a time with the other boundary DOFs fixed), one column for each static constraint mode.

Φ_m = Fixed interface dynamic vibration modes with boundary DOFs fixed, one column for each fixed interface dynamic vibration mode.

The boundary DOFs must be selected such that the original analysis model with the boundary DOFs fixed (i.e., with $\mathbf{u}_b = 0$) has no rigid body modes, that is, that the matrix \mathbf{K}_{ii} is positive definite.

- The first step in the CMS method is the calculation of the static constraint modes matrix Φ_c . One static solution is performed for each column in Φ_c .
- The second step is the calculation of the q fixed interface dynamic vibration eigenvalues

$$\Lambda_m = \begin{bmatrix} \lambda_1^m & & \\ & \dots & \\ & & \lambda_q^m \end{bmatrix} \text{ and eigenvectors } \Phi_m \text{ of the system}$$

(\mathbf{K}_{ii} , \mathbf{M}_{ii}). This calculation is performed using either the Bathe subspace iteration method (FREQUENCIES ... METHOD=CMS) or the Lanczos iteration method (FREQUENCIES ... METHOD=CMS-LANCZOS). The number of starting vectors for the Bathe subspace iteration method for the eigensolution of the system (\mathbf{K}_{ii} , \mathbf{M}_{ii}) is the maximum of $(2q, q + 8)$.

- The third step is the calculation of the reduced matrices \mathbf{K}_r and \mathbf{M}_r using

$$\begin{aligned} \mathbf{K}_r &= \Psi^T \mathbf{K} \Psi \\ \mathbf{M}_r &= \Psi^T \mathbf{M} \Psi \end{aligned}$$

in which

$$\Psi = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \Phi_c & \Phi_m \end{bmatrix}$$

Performing the indicated multiplications gives

$$\mathbf{K}_r = \begin{bmatrix} \mathbf{K}_{bb} + \mathbf{K}_{bi}\Phi_c & \mathbf{0} \\ \mathbf{0} & \Lambda_m \end{bmatrix}$$

and

$$\mathbf{M}_r = \begin{bmatrix} \mathbf{M}_{bb} + \mathbf{M}_{bi}\Phi_c + \Phi_c^T \mathbf{M}_{ib} + \Phi_c^T \mathbf{M}_{ii} \Phi_c & \mathbf{M}_{bi}\Phi_m + \Phi_c^T \mathbf{M}_{ii} \Phi_m \\ \text{symm.} & \mathbf{I} \end{bmatrix}$$

Notice that the boundary DOFs and the fixed interface dynamic vibration modes are coupled in the reduced mass matrix, \mathbf{M}_r . This coupling is important for CMS substructuring analysis. If there was no coupling in the reduced mass matrix, boundary DOFs accelerations would not excite the fixed interface dynamic vibration modes of the substructure.

- In the fourth step, there is an option to either perform a Rayleigh Ritz analysis (FREQUENCIES ... CMS-SUBSPACE=RITZ) or to perform the full Bathe subspace iteration method (FREQUENCIES ... CMS-SUBSPACE=SUBSPACE) to approximately calculate the lowest frequencies/modes of the original analysis model. In both options, the approximation is improved when the number of fixed interface dynamic vibration modes, q , is increased.

- When a Rayleigh-Ritz analysis is performed in the fourth step of the CMS method, the NEIGEN parameter is ignored. The number of frequencies/modes to be solved is always $p = k + q$.

When a Rayleigh-Ritz analysis is performed, the number of frequencies/modes to be solved is $p = k + q$, where k is the number of static constraint modes (number of boundary DOFs) and q is the number of fixed interface dynamic vibration modes. In this analysis, only one step of the Bathe subspace iteration method

is performed on the original analysis model (\mathbf{K} , \mathbf{M}), in which the starting vectors are taken from the columns of Ψ .

When the full Bathe subspace iteration method is performed, the number of frequencies/modes, p , to be solved is specified by the user, and can be much smaller than the number of static constraint modes, k , plus the number of fixed interface dynamic vibration modes, q , that is, it is allowed that $p \ll k + q$. In this analysis, first the full Bathe subspace method is performed to calculate the lowest p eigenvalues and eigenvectors of the system of reduced matrices (\mathbf{K}_r , \mathbf{M}_r), where the subspace iterations are continued until convergence is reached, then the p eigenvectors of the original analysis model are calculated using:

$$\mathbf{X}_1 = \Psi \mathbf{Q}_r$$

Modeling

- The user must specify the boundary DOFs using the DOF-ACTIVE command, and the number of fixed interface dynamic vibration modes using FREQUENCIES ... NFREQ-CMS. The number of boundary DOFs gives the number of static constraint modes, k .
- When FREQUENCIES ... AVL-OUT=TEXT, the program outputs the reduced system matrices \mathbf{K}_r and \mathbf{M}_r into a file with name *.out4. The .out4 files can then be imported into a 1D multibody system program for additional dynamic calculations, e.g., AVL EXCITE.
- When FREQUENCIES ... AVL-OUT=TEXT, the program also outputs the reduced system matrices \mathbf{K}_r and \mathbf{M}_r into a file with name *.genl. The .genl files can be read into the linear general element stiffness and mass matrices by using MATRIX STIFFNESS/MASS ... OPTION=FILE, for additional CMS substructuring calculations in ADINA.
- The reduced system matrices output to the *.out4 and *.genl files are always output in the form shown in the third step of the

CMS method above. That is, the boundary DOFs are output first, then the fixed interface dynamic vibration modes are output. The boundary DOFs are output in ascending order of node labels and DOFs, irrespective of the order in the DOF-ACTIVE command. For example, if FREQUENCIES ... NFREQ-CMS=3 and:

DOF-ACTIVE

2 Y-TRANSLATION

1 X-TRANSLATION

1 Y-TRANSLATION

2 X-TRANSLATION

The reduced matrices are output in the order:

1 X-TRANSLATION

1 Y-TRANSLATION

2 X-TRANSLATION

2 Y-TRANSLATION

Fixed interface dynamic vibration mode 1

Fixed interface dynamic vibration mode 2

Fixed interface dynamic vibration mode 3

- When the full Bathe subspace iteration method is performed in the fourth step of the CMS method, the user must specify the number of frequencies/modes, p , to be calculated using FREQUENCIES ... NEIGEN. It is allowed to input FREQUENCIES ... NEIGEN=0, in which case the fourth step is not performed, and ADINA only outputs the reduced system matrices \mathbf{K}_r and \mathbf{M}_r , provided FREQUENCIES ... AVL-OUT=TEXT. This is useful when only the reduced system matrices are needed.

- When a Rayleigh-Ritz analysis is performed in the fourth step of the CMS method, the NEIGEN parameter is ignored. The number of frequencies/modes to be solved is always $p = k + q$.
- The CMS method can be used for static, dynamic, or frequency substructuring analysis. In CMS substructuring analysis, the complete structure is split into a main structure and individual substructures, or components, where the substructures are connected to the main structure at the boundary DOFs. The main structure can undergo a nonlinear response, including large displacements/large strains, material nonlinearities, and contact nonlinearities, but the substructures can only undergo a linear response.

CMS substructuring analysis requires two runs. In the first run, the CMS method is used to compute the reduced system matrices, \mathbf{K}_r and \mathbf{M}_r , of each individual substructure. In the second run, the complete structure is assembled by connecting the substructures to the main structure. The substructures are modelled as linear general elements, where the reduced system matrices computed in the first run are read into the general element stiffness and mass matrices. The general element nodal connectivity must be consistent with the form output in the reduced system matrices. A static, dynamic, or frequency analysis of the complete structure is then performed. If the geometry and material properties of the substructures do not change, multiple second runs can be performed, each time reading in the same reduced system matrices.

The internal DOFs of the substructure are represented by the static constraint modes and fixed interface dynamic vibration modes of the substructure. In this sense, the static and dynamic modes can be considered as special interpolation functions of the substructure.

If the substructure undergoes a static response, no fixed interface dynamic vibration modes need to be calculated in the first run, i.e., FREQUENCIES ... NFREQ-CMS=0.

If the substructure undergoes a dynamic response, or if the frequencies/modes of the complete structure are to be computed, fixed interface dynamic vibration modes must be used to represent the dynamic response of the substructure, that is, FREQUENCIES ... NFREQ-CMS>0 in the first run. In the second run, the general element must have additional “dummy” nodes for the fixed

interface dynamic vibration modes of the substructure. For example, if the substructure has 6 dynamic modes, the general element must have 1 additional dummy node with 6 DOF. The dummy nodes should not be constrained nor connected to any element in the main structure. Rayleigh damping can be applied to the general elements to model damping in the substructure.

User hints

- It is allowed for the number of fixed interface dynamic vibration modes to be zero, $q = 0$. This is useful for CMS substructuring analysis, where the substructure only undergoes a linear static response (such that the static constraint modes fully represent the response of the substructure).
- It is assumed that the original analysis model with the boundary DOFs fixed (\mathbf{K}_{ii} , \mathbf{M}_{ii}) has no rigid body modes.
- If the original analysis model has rigid body modes with the boundary DOFs free, then a rigid body mode shift is required in the calculation of frequencies/modes of the original analysis model in the fourth step of the CMS method. This rigid body mode shift is specified as for other frequency analyses, see Section 6.2.
- If the structure is nonlinear and is preloaded, the CMS frequency analysis requires two runs, as for other frequency analyses, see Section 6.2.

This page intentionally left blank

12. Heat transfer capabilities in the ADINA program

- Most of the ADINA Thermal heat transfer capabilities are also included in the ADINA program. This chapter documents the supported analysis options of these capabilities in ADINA, and describes the features available for modeling.
- More details of the theory and modeling of heat transfer and TMC analysis can be found in the ADINA Thermal Theory and Modeling Guide, and the commands available are documented in the AUI Command Reference Manual, Volume 1.
- **Solutions of heat flow equations**

The following methods are available in ADINA TMC to solve heat flow equations:

- ▶ Euler backward method
- ▶ Euler forward method (for heat flow only solutions)
- ▶ Trapezoidal rule
- ▶ Bathe method

The Bathe method and the Euler backward method are commonly used in solutions of heat flow equations, with the Bathe method being the default method.

- **Bathe method**

The governing equation for heat flow solution can be written as follows:

$${}^{t+\Delta t}\mathbf{K} {}^{t+\Delta t}\Theta = {}^{t+\Delta t}\mathbf{Q} - \mathbf{C} {}^{t+\Delta t}\dot{\Theta} \quad (12-1)$$

where ${}^{t+\Delta t}\mathbf{K}$ is a heat flow conductivity matrix, \mathbf{C} is a heat flow capacity matrix, ${}^{t+\Delta t}\mathbf{Q}$ is an external heat load and ${}^{t+\Delta t}\Theta$ is a temperature vector at time $t + \Delta t$.

The method uses a two-step solution. The first sub-step, at time $t + \gamma\Delta t$ is solved using the Euler backward method, where γ is a

factor and $0 < \gamma \leq 1$ with the default value set to 0.5.

At time $t + \Delta t$, we use the following scheme:

$${}^{t+\Delta t}\dot{\Theta} = c_1 {}^t\Theta + c_2 {}^{t+\gamma\Delta t}\Theta + c_3 {}^{t+\Delta t}\Theta$$

where

$$c_1 = \frac{1-\gamma}{\gamma\Delta t} \quad c_2 = \frac{-1}{\gamma(1-\gamma)\Delta t} \quad c_3 = \frac{2-\gamma}{(1-\gamma)\Delta t}$$

For nonlinear solutions, ${}^{t+\Delta t}\Theta^{(i)}$ is equal to:

$${}^{t+\Delta t}\Theta^{(i)} = {}^{t+\Delta t}\Theta^{(i-1)} + \Delta\Theta^{(i)}, \quad {}^{t+\Delta t}\Theta^{(0)} = {}^t\Theta$$

then equation (12-1) can be presented as follows:

$${}^{t+\Delta t}\hat{\mathbf{K}}^{(i)}\Delta\Theta^{(i)} = {}^{t+\Delta t}\mathbf{Q} - \mathbf{C}\left(c_1 {}^t\Theta + c_2 {}^{t+\gamma\Delta t}\Theta + c_3 {}^{t+\Delta t}\Theta^{(i-1)}\right) - {}^{t+\Delta t}\Theta^{(i-1)} {}^{t+\Delta t}\mathbf{K}^{(i-1)} \quad (12-2)$$

where

$${}^{t+\Delta t}\hat{\mathbf{K}}^{(i)} = {}^{t+\Delta t}\mathbf{K}^{(i)} + c_3\mathbf{C}$$

12.1 Analysis options

- Three analysis options are available:
 - ▶ heat transfer analysis only
 - ▶ one-way heat flow coupled to mechanical analysis
 - ▶ fully thermo-mechanical-coupled (TMC) analysis (iterative or direct option)
- The first analysis type involves only heat transfer with no structural equations or coupling. Results from this analysis can be later used in a structural solution, e.g., via the restart option.

- The second analysis type is a one-way coupling, when the thermal solution affects the structural solution, but the structural solution does not affect the thermal solution.

In this case, the heat transfer and structural equations are solved separately, with the temperatures from the heat transfer passed back to the structural problem for calculation of thermal expansion and temperature-dependent material properties.

- The third analysis type is iterative or direct thermo-mechanical coupling which is a two-way coupling where both the thermal and structural solutions are interdependent.

In this case, the thermal solution can affect the structural solution and the structural solution can affect the thermal solution.

This option should be used for nonlinear thermos-mechanical solutions only.

- In the case of iterative TMC, at the beginning of each time step, the structural model is solved for the displacements using the current temperatures. Then the heat transfer model is solved for the temperatures using the current displacements. This cycle constitutes one TMC equilibrium iteration. TMC convergence is then assessed, and if it is not reached, then the structural and heat transfer models are solved again using the new current displacements and new current temperatures. This process is repeated until TMC convergence is reached. Note that within each TMC equilibrium iteration, the heat transfer and structural models each have their own internal iteration procedure and convergence criteria.

- The same TMC convergence parameter is used in the displacement and temperature convergence checks.

- The temperature convergence is checked as follows:

$$\frac{\left\| {}^{t+\Delta t}T^{(i)} - {}^{t+\Delta t}T^{(i-1)} \right\|_2}{\left\| {}^{t+\Delta t}T^{(0)} \right\|_2} < \text{TOLL} \quad (12.1)$$

- The displacement convergence is checked as follows:

$$\frac{\left\| {}^{t+\Delta t}\mathbf{U}^{(i)} - {}^{t+\Delta t}\mathbf{U}^{(i-1)} \right\|_2}{\left\| {}^{t+\Delta t}\mathbf{U}^{(i)} \right\|_2} < \text{TOLL} \quad (12.2)$$

where i denotes the TMC iteration.

- For the direct TMC option, the program solves one system of equations which includes displacement and temperature increments and there is no TMC iteration process; only the standard Newton-Raphson iteration procedure is applied for which convergence or displacements/forces and temperatures are checked separately using equations (12.1) and (12.2) respectively.

- In strongly coupled problems, a temperature relaxation factor can be used to help reach convergence. This parameter defaults to 1.0, which corresponds to no relaxation.

The temperatures used in the structural analysis in the case of temperature relaxation at a TMC iteration k are based on the temperatures in the last heat transfer TMC iteration $k-1$ as well as the prior heat transfer TMC iteration $k-2$.

$$\theta_{structure}^{(k)} = (1 - \lambda_1) \theta_{heat}^{(k-2)} + \lambda_1 \theta_{heat}^{(k-1)}$$

where λ_1 is the temperature relaxation factor.

- Note that decreasing the relaxation factor usually reduces the chances of an oscillating solution, but if decreased too much will also slow down convergence.
- The following analysis options, available in ADINA-T, are **not** available in the ADINA program for heat transfer solutions:
 - thermal eigenvalue calculations
 - seepage
 - analogous field problems
 - Joule heating

12.2 Modeling

- One finite element model is used for mechanical and heat flow

solutions with the following features:

- ▶ Element groups have two different sets of material models, one mechanical and the other one, thermal (called in the AUI input TMC-MATERIAL).
- ▶ The same element integration order is used for mechanical and thermal solutions.
- ▶ Element birth/death times are the same for mechanical and thermal solutions.
- For higher order elements used in the displacement and stress solutions, the ADINA program has an option to calculate temperatures using corner nodes only (TMC-CONTROL TEMP-INTERP=CORNER).
- Different time steps can be used for temperature and displacement solutions in case of the one way coupling option. For the fully coupled solution, the same time step is used.
- Steady and transient heat transfer solution can be used with either statics or implicit dynamics structural solution.
- No temperature tape is used during heat flow analysis in the ADINA program.
- Different time spans can be used for thermal and mechanical solutions for one way coupling. If the thermal time span is shorter than the mechanical time span, then the last calculated temperature field is used.
- Different solution starting times can be used for thermal and mechanical solutions for one way coupling. For example, first thermal solution can be completed and then mechanical solution can follow with the solution starting time larger than the thermal solution ending time. The last obtained temperature field will then be used in the mechanical solution.
- Explicit dynamics cannot be coupled to heat transfer analysis.

- For shell elements, two degrees of freedom are present at each node, one corresponding to the top surface and one to the bottom surface.
- For shell elements, heat flow loads can be applied either on top, bottom or mid-surface. Note that different temperature loads can be applied to the top and bottom surface. When temperatures are applied to the midsurface they are assigned to both top and bottom shell surfaces.
- Single and layered shells can be used in TMC solutions.
- Automatic time stepping (ATS) can be used to vary the time step size in order to help obtain a converged solution. If no convergence is obtained in the original step, the program automatically subdivides the time steps, and attempts to solve again. Further subdivision can be done until convergence is reached or the time step size becomes smaller than a minimum value. This minimum value is set as the original time step divided by a user-specified scaling factor.
- This automatic time stepping procedure is used in the solution of heat transfer analyses and one-way coupled analyses. For iteratively coupled TMC analyses the structural ATS procedure of Section 7.2.4 is used instead. Note that the structural ATS procedure has many more features, and is better suited for nonlinear problems involving contact, geometric and material structural nonlinearities. For direct coupled TMC, the structural ATS procedure is applied.
- Line search can also be used in the heat transfer solution in both heat transfer only and coupled heat transfer analyses.
- Mesh glueing can be used for temperature DOFs for 2-D and 3-D solid elements. If the GLUEMESH command is specified in the input and a TMC analysis is requested, then, in addition to displacement DOFs, temperature DOFs are also constrained between glued domains. If heat flow only option is used, then only temperature DOFs are constrained.

12.3 Soil consolidation analysis

- Coupled diffusion-stress analysis problems can be solved with the ADINA system. Such is the phenomenon of soil consolidation, in which a soil under load settles due to the dissipation of excess internal fluid pressure.
- This chapter discusses the solution of soil consolidation problems using ADINA-TMC. However, it is recommended that soil consolidation problems be solved instead using the porous media formulation of ADINA, see Section 3.12.
- In ADINA-TMC, the soil consolidation modeling capability is based on the linear consolidation theory, as described below.
- The soil is regarded here as a porous medium, consisting of a solid skeleton (soil particles in contact) and of interconnected voids partially or totally filled with fluid. The soil porosity, n , is defined as the proportion of voids per unit total volume. The void ratio, e , is defined as the proportion of voids to solids in any given volume. The fluid content, θ , at a given fluid pressure π , is defined as the change in fluid volume, per unit total volume, between the strained state corresponding to π and a reference unstrained state.
- The linear consolidation theory provides a macroscopic description of the soil response, based on the following assumptions:
 - ▶ The soil skeleton behavior is linear elastic isotropic.
 - ▶ The fluid is incompressible.
 - ▶ The fluid flows through the porous soil according to Darcy's law:

$$\mathbf{v} = -\mathbf{k} \cdot \nabla \pi \quad (12.1)$$

where

\mathbf{v} = fluid velocity vector
 \mathbf{k} = soil permeability matrix
 π = fluid pressure

- Considering only small (macroscopic) strains in the soil and small velocities in the fluid, a linear stress-strain relation can be derived:

$$\boldsymbol{\sigma} = \mathbf{C}\mathbf{e} - \alpha\pi\mathbf{1} \quad (12.2)$$

where

- $\boldsymbol{\sigma}$ = macroscopic stress tensor
- \mathbf{e} = macroscopic strain tensor
- \mathbf{C} = macroscopic stress-strain law matrix of the soil skeleton
- α = first soil consolidation parameter
- $\mathbf{1}$ = Kronecker delta vector

- It is also assumed that the fluid content θ varies linearly with the fluid pressure and the soil volumetric strain, i.e.

$$\theta = \alpha e_v + \beta\pi \quad (12.3)$$

where

- e_v = soil skeleton volumetric strain
- β = second soil consolidation parameter

- The general equations governing transient soil consolidation can then be established. First, the macroscopic stresses defined in equation (12.2) must satisfy the equilibrium condition:

$$\nabla \cdot \boldsymbol{\sigma} + \mathbf{f}^b = \mathbf{0} \quad (12.4)$$

where \mathbf{f}^b are body forces. Second, the continuity condition for incompressible fluid flows:

$$\frac{\partial \theta}{\partial t} - q^b = -\nabla \cdot \mathbf{v} \quad (12.5)$$

where q^b is the internal fluid flow generation, used together with Darcy's equation (12.1), and with equation (12.3), yields the diffusion equation controlling the variation of the fluid pressure:

$$\nabla \cdot (\mathbf{k} \cdot \nabla \pi) = \alpha \frac{\partial e_v}{\partial t} + \beta \frac{\partial \pi}{\partial t} - q^b \quad (12.6)$$

We can now see that the stress equation (12.4) and the diffusion equation (12.6) constitute a coupled diffusion-stress equation system.

- Note on the soil consolidation parameters: From equation (12.6), it can be seen that the first soil consolidation parameter α is a ratio between the outward flux of fluid out of the soil and the variation in the soil volume, whereas the second soil consolidation parameter β is a ratio between the outward flux of fluid and the variation in the fluid pressure.

- yield the following equation:

$$\begin{bmatrix} 0 & 0 \\ \mathbf{K}_{\pi u}^T & \mathbf{M} \end{bmatrix} \begin{bmatrix} \frac{\partial \mathbf{u}}{\partial t} \\ \frac{\partial \pi}{\partial t} \end{bmatrix} + \begin{bmatrix} \mathbf{K}_u & \mathbf{K}_{u\pi} \\ 0 & \mathbf{K}_\pi \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \pi \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{q} \end{bmatrix} \quad (12.7)$$

where

$$\mathbf{K}_u = \int_V \mathbf{B}_u^T \mathbf{C} \mathbf{B}_u dV$$

$$\mathbf{K}_{u\pi} = - \int_V \mathbf{B}_u^T \alpha \mathbf{H} dV$$

$$\mathbf{K}_{\pi u}^T = - \int_V \mathbf{H}^T \alpha \mathbf{B}_u dV$$

$$\mathbf{K}_\pi = \int_V \mathbf{B}_\pi^T \mathbf{k} \mathbf{B}_\pi dV$$

$$\mathbf{M} = \int_V \mathbf{H}^T \beta \mathbf{H} dV$$

\mathbf{f}, \mathbf{q} = load vectors (forces, fluid flows)

\mathbf{H} = interpolation matrix

$\mathbf{B}_u, \mathbf{B}_\pi$ = gradient matrices for the displacements \mathbf{u} and

pressure π

- Equation (12.7) can be written as follows:

$$\mathbf{K}_u \mathbf{u} = \mathbf{f} - \mathbf{K}_{u\pi} \pi \quad (12.8)$$

$$\mathbf{M} \frac{\partial \pi}{\partial t} + \mathbf{K}_\pi \pi = \mathbf{q} - \mathbf{K}_{\pi u}^T \frac{\partial \mathbf{u}}{\partial t} \quad (12.9)$$

Equation (12.8) is solved using the ADINA model, for a given fluid pressure field, and equation (12.9) is solved, for a given displacement field. An iterative procedure can therefore be used to solve this equation system.

- UCMAT3C, TUSR2C and/or TUSR3C, should be used in soil consolidation analysis. Note that, in TUSR2C and TUSR3C, KEY=5 provides the first soil consolidation parameter α .

12.4 Piezoelectric analysis

- The solution of piezoelectric problems (direct and converse effects) can be performed with ADINA-TMC. Similar coupled mechanical/electrical problems including magnetostrictive, electrostrictive materials, and shape memory alloys, can also be solved with ADINA-TMC.
- In a piezoelectric material, the stresses depend on the electric potential, and the electric potential depends on the stresses. Hence, the following system must be solved:

$$\begin{bmatrix} \mathbf{K}_{UU} & \mathbf{K}_{U\Phi} \\ \mathbf{K}_{\Phi U} & \mathbf{K}_{\Phi\Phi} \end{bmatrix} \begin{bmatrix} \mathbf{U} \\ \Phi \end{bmatrix} = \begin{bmatrix} \mathbf{F}_U^B + \mathbf{F}_U^S \\ \mathbf{F}_\Phi^S \end{bmatrix} \quad (12.10)$$

where

\mathbf{K}_{UU} = mechanical stiffness

$\mathbf{K}_{\Phi\Phi}$ = electric permittivity matrix

$\mathbf{K}_{U\phi}$ = piezoelectric matrix

$$\mathbf{K}_{\phi U} = \mathbf{K}_{U\phi}^T$$

\mathbf{F}_U^B = body force loading vector

\mathbf{F}_U^S = surface force loading vector

\mathbf{F}_ϕ^S = surface charge density vector

\mathbf{U} = vector of structural displacements

Φ = vector of electric potentials

- For solution with ADINA-TMC, the equation system (12.10) above is written as follows:

$$\mathbf{K}_{UU} \mathbf{U} = \mathbf{F}_U^B + \mathbf{F}_U^S - \mathbf{K}_{U\phi} \Phi \quad (12.11)$$

$$\mathbf{K}_{\phi\phi} \Phi = \mathbf{F}_\phi^S - \mathbf{K}_{\phi U} \mathbf{U} \quad (12.12)$$

- Equation (12.11) can be solved using the ADINA model, assuming that the electric potential field is known, and equation (12.12) can be solved using the same model, assuming that the displacements in the structure are known.
- The ADINA user-supplied subroutines UCMAT2P and/or UCMAT3P, and the user-supplied subroutines TUSR2P and/or TUSR3P, should be used in piezoelectric analysis.
- The piezoelectric solution option can also be used in FSI analysis.
- More information about the theory and applications of piezoelectricity with the finite element method can be found in the following references:

ref. Gaudenzi P., Bathe K. J., "Recent Applications of an Iterative Finite Element Procedure for the Analysis of Electrostatic Materials," *Proceedings of the 4th International Conference on Adaptive Structures*, Nov. 2-4, Köln, Germany, 1993.

ref. Tzou H. S., Tseng C. I., "Distributed Piezoelectric

Sensor/Actuator Design for Dynamic
Measurement/Control of Distributed Parameter Systems:
A Piezoelectric Finite Element Approach," *Journal of
Sound and Vibration*, Vol. 138 (1), pp. 17-34, 1990.

12.5 Thermal coupling between solids and fluids

ADINA provides two solution options for thermo-fluid-structure interaction problems:

One-way thermal coupling: The temperature field is solved entirely in the CFD model, where solid elements are modeled as fluid elements with fluid velocity and pressure set to zero i.e., only temperature DOFs are active in the “solid” part. The solid temperature field is then mapped to the solid model for every FSI iteration. This method does not account for any heat generation in the solid.

Two-way (and one-way) thermal coupling between solids and fluids: The energy equation is solved separately in the fluid model and the solid model, and the domains are coupled together by enforcing the same temperature and heat flux at the fluid-structure interface. This method is known as thermo-FSI (or TFSI) analysis. The benefits are:

- ▶ heat generation in the solid model (e.g. due to frictional heating, plasticity, viscous losses, etc.) affects the temperature in the fluid model
 - ▶ no need for temperature mapping
 - ▶ no need to create a solid mesh in the fluid model
- Note that the BTFSI option in version 9.0 has been replaced by the above standard TFSI option. If a user specifies in the input the BTFSI option (formerly known as TQ and QT), the program uses the standard TFSI solution procedure.

Figures 12.5-1 and 12.5-2 below illustrate the difference between the one- and two-way thermal coupling methods.

- In TFSI:
 - ▶ Two-way thermal coupling (TFSI) is only supported for FCBI-C elements with iterative FSI coupling. It is not

supported for direct FSI coupling.

- ▶ All thermo-mechanical coupling (TMC) options for heat flow solution in the solid model are available. That is, one-way TMC coupling, iterative TMC coupling, and direct TMC coupling methods can all be used in the solid model.
- ▶ Iso-beam and shell elements account for the through-thickness temperature gradient due to different temperatures on the top and bottom surfaces of the element. However, the beam element assumes uniform through-thickness temperature equal to the average temperature between the top and bottom surfaces.

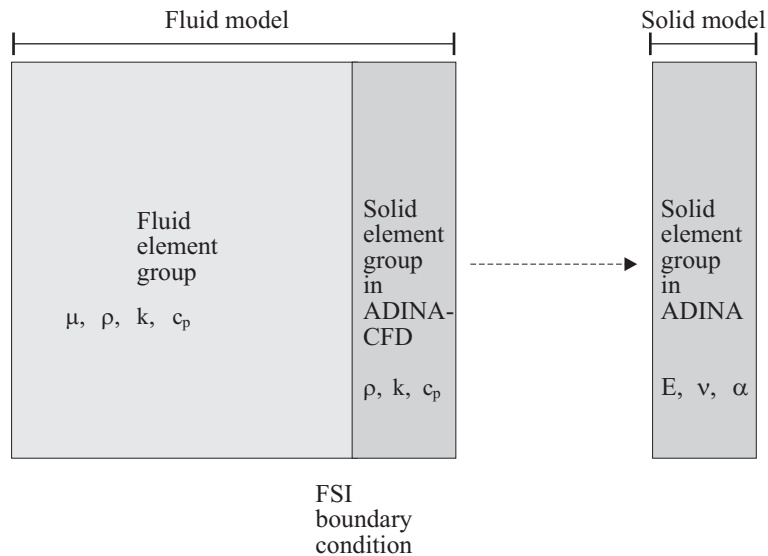


Figure 12.5-1: One-way thermal coupling (conjugate heat transfer)

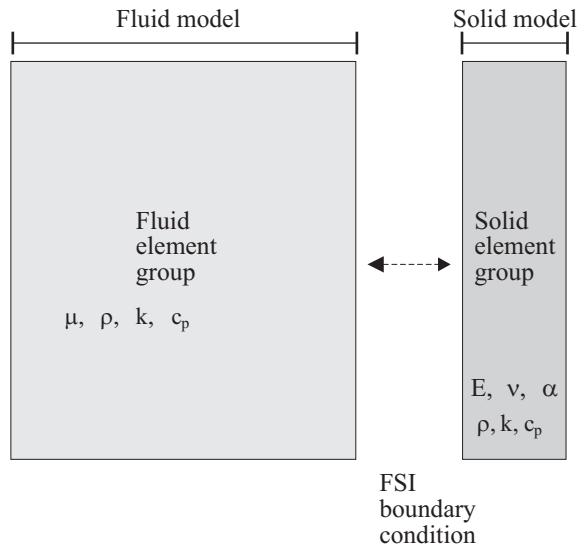


Figure 12.5-2: Two-way thermal coupling (TFSI)

13. Postprocessing considerations

- In this chapter we discuss those aspects of the AUI (ADINA User Interface) which require an in-depth discussion.
- This chapter discusses the postprocessing of results from all five solution programs ADINA, ADINA-T, ADINA-F, ADINA-FSI, ADINA-TMC.

13.1 Calculation of results within the AUI

- The ADINA system programs generate a wide variety of results, for example, stresses, displacements, reactions, heat fluxes, contact segment tractions. You access all of these results using several quite general AUI commands. To understand the inputs to these commands, the following concept is important to remember.
- The AUI must know the answers to the following four questions before it can obtain a result.
 - ▶ What result is to be obtained?
 - ▶ Where (in the model) is the result to be evaluated?
 - ▶ When (for which load step, mode shape, etc) is the result to be evaluated?
 - ▶ How is the result to be evaluated?

You can supply answers to each of these questions within the AUI. "What" is controlled by variables. "Where" is controlled by model points, model combination points, model lines, zones and result grids. "When" is controlled by response depictions and response-range depictions. "How" is controlled by smoothing depictions and result control depictions.

- It is also important to understand that the AUI can transform results computed by the solution programs (the available results) to the results that you request (the requested results). The AUI can make the following transformations:

Spatial transformations:

- ▶ The AUI can interpolate results available at the element integration points to other points within the elements, such as the element local nodes.
- ▶ The AUI can combine the results from all of the elements attached to a node into a single result (smoothing).
- ▶ The AUI can interpolate the results from the nodes to points within elements.

Time transformations:

- ▶ The AUI can interpolate results available at two solution times to solution times between these two times.

Resultant transformations:

- ▶ The AUI can combine results at a single point into a single result. For example, the AUI can compute the effective stress at a point from the stress components at the point.

Combination transformations:

- ▶ The AUI can combine results according to response spectrum formulas, harmonic and random vibration formulas or user-supplied combinations (known as response-combinations). The AUI can also scan for the most extreme result for given solution times and return the most extreme value (envelope combinations).

These transformations are organized in a hierarchy such that each transformation can request information from the transformation below it. The hierarchy is described in the following examples.

Example 1: As an example involving only the spatial transformation, consider the evaluation of a smoothed stress component at a point within an element (not necessarily a point at which the results were computed by ADINA) for an available time

(a time for which results were computed by ADINA). This stress component is calculated by the spatial transformation as follows. The spatial transformation interpolates the available stress component data from the integration points to the nodes, performs smoothing, then interpolates the stress component to the requested point.

Example 2: As an example involving only the time and spatial transformations, consider the evaluation of a smoothed stress component at a point within an element for a requested time (not necessarily a time for which results were computed by ADINA). This stress component is calculated by the time transformation. The time transformation requests the stress component for the available solution times nearest to the requested time, then interpolates in time to the requested solution time. The stress components at the available solution times are calculated by the spatial transformation as in Example 1.

In the above example, the spatial transformation acts as a subroutine that is called by the time transformation.

Example 3: As an example involving the resultant, time and spatial transformations, consider the evaluation of smoothed effective stress at a point within an element for a requested time. The effective stress is computed by the resultant transformation. The resultant transformation requests the stress components from the time transformation, then computes the effective stress. The time transformation computes the stress components as in Example 2.

In the above example, the time transformation acts as a subroutine that is called by the resultant transformation.

Example 4: As an example involving the combination, time and spatial transformations, consider the evaluation of stress component difference at a point within an element between two requested times. (E.g., $\sigma_{xx}(\text{time } 2.0) - \sigma_{xx}(\text{time } 1.0)$.) The stress component difference is computed by the combination transformation. The combination transformation requests the stress components from the time transformation, then computes the difference. The time transformation computes the stress components as in Example 2.

In the above example, the time transformation acts as a subroutine that is called by the combination transformation.

- In the above examples, the AUI automatically determines the hierarchy order. However, when considering examples involving both the resultant and the combination transformations, you can control the order of these transformations. There are two cases:

Resultants before combinations: The combination transformation can request results from the resultant transformation, the resultant transformation can request results from the time transformation, the time transformation can request results from the spatial transformation.

In other words, the combination transformation acts on resultants.

Resultants after combinations: The resultant transformation can request results from the combination transformation, the combination transformation can request results from the time transformation, the time transformation can request results from the spatial transformation.

In other words, the combination transformation acts on the variables contained within the resultant.

The default is “resultants before combinations”.

The following examples show cases in which the hierarchy order is important.

Example 5: Consider the calculation of the radial stress in response spectrum analysis. The response spectrum calculation is a combination transformation and the radial stress calculation is a resultant transformation.

If “resultants before combinations” is used, the combination transformation requests the radial stresses for each mode shape from the resultant transformation, then the combination transformation combines the modal radial stresses using the response spectrum formulas.

If “resultants after combinations is used”, the resultant transformation requests the stress components from the combination transformation, then the resultant transformation computes the radial stress. The combination transformation calculates the stress components using the modal stress components

and the response spectrum formulas.

In this example, “resultants before combinations” is correct, and it would be incorrect to use “resultants after combinations”.

Example 6: Consider the calculation of the effective stress for a response combination involving the addition of two load cases. The response combination calculation is a combination transformation and the effective stress calculation is a resultant transformation.

If “resultants before combinations” is used, the combination transformation requests the effective stresses for each load case from the resultant transformation, then the combination transformation adds the effective stresses.

If “resultants after combinations is used”, the resultant transformation requests the stress components from the combination transformation, then the resultant transformation computes the effective stress. The combination transformation adds the stress components from the two load cases.

In this example, “resultants after combinations” is correct, and it would be incorrect to use “resultants before combinations”.

Example 7: Consider the calculation of the effective stress in harmonic vibration analysis, in which you enter the value of ωt . The harmonic vibration analysis calculation is a combination transformation and the effective stress calculation is a resultant transformation.

If “resultants before combinations” is used, the combination transformation requests the effective stresses for each mode shape from the resultant transformation, then the combination transformation combines the modal effective stresses using the harmonic vibration formulas.

If “resultants after combinations” is used, the resultant transformation requests the stress components from the combination transformation, then the resultant transformation computes the effective stress. The combination transformation calculates the stress components using the modal stress components and the harmonic vibration formulas.

In this example, “resultants after combinations” is correct, and it would be incorrect to use “resultants before combinations”.

Example 8: Consider the calculation of the resultant $1 + u_i$, where

u_i is a displacement component, in harmonic vibration analysis.

If “resultants before combinations” is used, the combination transformation requests the quantity $1 + \phi$ for each mode shape from the resultant transformation (where ϕ is the corresponding eigenvector component for each mode shape), then the combination transformation combines the quantities $1 + \phi$ using the harmonic vibration formulas.

If “resultants after combinations” is used, the resultant transformation requests u_i from the combination transformation, then the resultant transformation computes $1 + u_i$. The combination transformation calculates u_i using ϕ for each mode shape and the harmonic vibration formulas.

In this example, “resultants after combinations” is correct, and it would be incorrect to use “resultants before combinations”.

Example 9: Consider the calculation of effective stress in random vibration analysis.

If “resultants before combinations” is used, the combination transformation requests the effective stresses for each mode shape from the resultant transformation, then the combination transformation combines the modal effective stresses using the random vibration formulas.

If “resultants after combinations” is used, the resultant transformation requests the stress components from the combination transformation, then the resultant transformation computes the effective stress. The combination transformation calculates the stress components using the modal stress components and the random vibration formulas.

In this example, neither “resultants before combinations” nor “resultants after combinations” is correct. It would be better if the AUI could estimate the effective stress by changing the signs of the stress components output from the combination transformation, so as to maximize the effective stress.

- The general rule of thumb is, if both the combination transformation and the resultant transformation are linear, it doesn't matter which rule to use; both will give the correct results. If the combination transformation is nonlinear and the resultant

transformation is linear, use “resultants before combinations”. If the combination transformation is linear and the resultant transformation is nonlinear, use “resultants after combinations”. If both transformations are nonlinear, then neither rule will give the correct results.

- There is an additional consideration when the model point is a combination point, for example a node combination point (in which the results at the nodes in the combination point are summed).

When “resultants before combinations” is used, the combination transformation requests the summed resultant (the summing being performed by the combination point).

When “resultants after combinations” is used, the summing (by the combination point) is performed on the variables referenced by the resultant, then the resultant is taken.

Example 10: Consider the calculation of the effective stress for a response combination involving the subtraction of two load cases. Here the effective stress is evaluated at a node combination point.

If “resultants before combinations” is used, the combination transformation requests the summed effective stresses for each load case, then the combination transformation subtracts the summed effective stresses.

If “resultants after combinations”, the combination point requests the effective stresses from the resultant transformation and then sums the effective stresses. The resultant transformation computes the effective stresses from the stress components. The combination transformation subtracts the stress components from the two load cases.

- In the following, we discuss what results are available, where they can be evaluated and how they are evaluated.

13.1.1 What results are available

- The results computed by the ADINA system programs are made available to the AUI when the corresponding porthole file is loaded. In many cases you control the results that are available when you define the model during preprocessing. For example, reactions are computed if you check the "Calculate Reaction

Forces/Moments" box within the Control→Miscellaneous Options dialog box.

Regarding results output along with the elements, in general you can either output stresses or forces, but not both. This choice is made within the element group definitions dialog box. For example, for a 2-D solid element group, the choice is made with the "Calculated Element Results" buttons.

When you choose to output stresses, the actual results output depend upon the material model and upon the stress reference system employed. These results are described in Chapter 2 of this manual for each element group.

Regarding results output along with the contact surfaces, you can output either segment tractions or forces. This choice is made within the contact group definitions dialog box. However, ADINA calculates segment tractions only for contactor contact surfaces. The actual results output are described in Chapter 4 of this manual.

- Each type of result output by ADINA is referred to by a variable name.

The variables can be classified according to where (in the model) they are output by the solution program and how they are interpolated within the model. The variable location types are:

Location-independent: Variables such as time or frequency that are not associated with specific points in the model.

Node field: Variables such as displacement, velocity or acceleration that are output at node points and that can be interpolated within elements (using the nodal shape functions).

Element/layer field: Variables such as stresses that are output within elements or element layers and that can be interpolated within elements. These variables are output by the solution program at the integration points.

Contact: Variables such as normal traction that are output at contact segments (ADINA only).

Draw bead: Variables such as restraining traction and uplifting traction that are output at draw bead segments (ADINA only).

Section field: Variables such as moments that are output within the sections of elements (midsurface of shells or neutral line of Hermitian beams) and that can be interpolated within element sections (ADINA only).

Radiation field: Variables that are output at radiosity segments.

Line contour: Variables that are output for each fracture mechanics line contour (ADINA only).

Virtual shift: Variables that are output for each fracture mechanics virtual shift (ADINA only).

Node discrete: Variables such as reaction that are output at node points and that cannot be interpolated within elements.

Element local node: Variables such as consistent nodal forces that are output at the element local nodes and that cannot be interpolated within elements.

Element local node field: Variables such as shell thicknesses that are output at the element local nodes and that can be interpolated within elements.

Coordinate: Variables such as coordinate.

Element face: Variables that are output at the element edges (2D analysis) or element faces (3D analysis) of ADINA-F elements.

Mesh surface: Variables, such as the surface normal, that can only be evaluated at mesh integration or mesh extreme points.

The variables can also be classified according to function. The classifications used in the AUI are: *displacement, velocity, acceleration, temperature, fluid, latent heat, stress, strain, force, traction, reaction, flux, electric field, fracture, failure criterion, eigenvector, prescribed load, coordinate, thickness, time, frequency/mode, contact, energy, miscellaneous, user-defined.*

Tables 13.1-1 to 13.1-25 show all of the variables. Each table shows all of the variables of the given function category. In the

tables, we compress the variable names using characters enclosed in parentheses. The compression rules are as follows:

Numerical lists: Entries enclosed in parentheses are to be expanded with one variable per character in the parentheses. For example, OVALIZATION- (123456) represents the six variables OVALIZATION-1, OVALIZATION-2, OVALIZATION-3, OVALIZATION-4, OVALIZATION-5, OVALIZATION-6.

Vectors: Entries enclosed in parentheses are to be expanded with one variable per character in the parentheses. For example, (ABC) -DISPLACEMENT represents the three variables A-DISPLACEMENT, B-DISPLACEMENT, C-DISPLACEMENT.

Symmetric tensors: Entries enclosed in parentheses are to be expanded to give components as in this example: STRESS- (XYZ) represents the six variables STRESS-XX, STRESS-YY, STRESS-ZZ, STRESS-XY, STRESS-XZ, STRESS-YZ.

Other cases: Entries enclosed in parentheses and separated by commas are to be expanded as in this example: CURVATURE- (RB, SB, RBSB, T) represents the four variables CURVATURE-RB, CURVATURE-SB, CURVATURE-RBSB, CURVATURE-T.

We also use the * to denote any number starting from 1, as in this example: TIME_FUNCTION_* represents the variables TIME_FUNCTION_1, TIME_FUNCTION_2, etc.

The tables give notes for those variables that are not self-explanatory.

Table 13.1-1: Displacement variables

Variable name	Variable location type	Notes
(123) -DISPLACEMENT	Node field	1
(ABC) -DISPLACEMENT	Node field	2
(XYZ) -DISPLACEMENT	Node field	
DISPLACEMENT_MAGNITUDE	Node field	3
FLUID_P0	Node field	
FLUID_POTENTIAL	Node field	
OVALIZATION- (123456)	Node field	
(123) -ROTATION	Node field	1
(ABC) -ROTATION	Node field	
(V1, V2) -ROTATION	Node field	
(XYZ) -ROTATION	Node field	
WARPING- (123456)	Node field	

1) These are the components of the displacements and rotations transformed by the AUI into a user-specified coordinate system. More information is given at the end of this section.

2) The A, B and C components refer to the skew coordinate system (if one is used). The V1, V2 components refer to the shell rotation directions (for shell midsurface nodes with five degrees of freedom). The X, Y and Z components always refer to the global coordinate system. The AUI can transform results that were calculated by ADINA in one coordinate system to another coordinate system, see Table 13.1-24.

This comment holds for velocities, accelerations and eigenvectors as well as displacements.

3) DISPLACEMENT_MAGNITUDE is the square root of the sum of the squares of the displacement components, as computed by the AUI.

Table 13.1-2: Velocity variables

Variable name	Variable location type	Notes
(123)-ANGULAR_VELOCITY	Node field	1
(ABC)-ANGULAR_VELOCITY	Node field	
(V1,V2)-ANGULAR_VELOCITY	Node field	
(XYZ)-ANGULAR_VELOCITY	Node field	
ELEMENT_(123)-VELOCITY	Element field	1
ELEMENT_(XYZ)-VELOCITY	Element field	
FLUID_P0_VELOCITY	Node field	2
FLUID_POTENTIAL_VELOCITY	Node field	3
NODAL_SLIP_VELOCITY	Node field	4
NODAL_SLIP_VELOCITY-(XYZ)	Node field	5
OVALIZATION_VELOCITY-(123456)	Node field	6
PORE_PRESSURE_VELOCITY	Node field	7
(123)-VELOCITY	Node field	1
(ABC)-VELOCITY	Node field	
(XYZ)-VELOCITY	Node field	
VELOCITY_MAGNITUDE	Node field	8
WARPING_VELOCITY-(123456)	Node field	9

1) These are the components of the velocities, angular velocities, etc., transformed by the AUI into a user-specified coordinate system. More information is given at the end of this section.

2) This is the 1st time derivative of the fluid P_0 degree of freedom.

3) This is the 1st time derivative of the fluid potential degree of freedom.

4) NODAL_SLIP_VELOCITY is the magnitude of the contact slip velocity. It is output when the “new” contact segments are used in frictional contact.

5) NODAL_SLIP_VELOCITY-(XYZ) are the components of the contact slip velocity. These are output when the “new” contact segments are used in frictional contact.

6) These are the 1st time derivatives of the ovalization degrees of freedom.

7) This is the 1st time derivative of the pore pressure (see Table 13.1-7 for the pore pressure).

8) `VELOCITY_MAGNITUDE` is the square root of the sum of the squares of the velocity components, as computed by the AUI.

9) These are the 1st time derivatives of the warping degrees of freedom.

Table 13.1-3: Acceleration variables

Variable name	Variable location type	Notes
(123)-ACCELERATION	Node field	1
(ABC)-ACCELERATION	Node field	
(XYZ)-ACCELERATION	Node field	
ACCELERATION_MAGNITUDE	Node field	2
(123)-ANGULAR_ACCELERATION	Node field	1
(ABC)-ANGULAR_ACCELERATION	Node field	
(V1,V2)-ANGULAR_ACCELERATION	Node field	
(XYZ)-ANGULAR_ACCELERATION	Node field	
FLUID_P0_ACCELERATION	Node field	3
FLUID_POTENTIAL_ACCELERATION	Node field	4
OVALIZATION_ACCELERATION-(123456)	Node field	5
PORE_PRESSURE_ACCELERATION	Node field	6
WARPING_ACCELERATION-(123456)	Node field	7

1) These are the components of the accelerations and angular accelerations transformed by the AUI into a user-specified coordinate system. More information is given at the end of this section.

2) `ACCELERATION_MAGNITUDE` is the square root of the sum of the squares of the acceleration components, as computed by the AUI.

3) This is the 2nd time derivative of the fluid P_0 degree of freedom.

4) This is the 2nd time derivative of the fluid potential degree of freedom.

5) These are the 2nd time derivatives of the ovalization degrees of freedom.

6) This is the 2nd time derivative of the pore pressure (see Table 13.1-7 for the pore pressure).

7) These are the 2nd time derivatives of the warping degrees of freedom.

Table 13.1-4: Temperature variables

Variable name	Variable type	Notes
ELEMENT_TEMPERATURE	Element field	1
TEMPERATURE	Node field	

1) This is the temperature as output at the element integration points.

Table 13.1-5: Fluid variables

Variable name	Variable location type	Notes
CELL_PECLET_NUMBER	Element field	
CELL_REYNOLDS_NUMBER	Element field	
EFFECTIVE_VISCOSITY	Element field	
MASS_RATIO_*	Node field	
NODAL_VISCOSITY	Node field	
OMEGA- (XY, XZ, YZ)	Element field	
STREAM_FUNCTION	Node field	
TOTAL_HEAD	Node field	
TURBULENCE_EPSILON	Node field	
TURBULENCE_K	Node field	
TURBULENCE_W	Node field	

Table 13.1-6: Latent heat variables

Variable name	Variable location type	Notes
LATENT_HEAT_INTERFACE_*	Node field	

Table 13.1-7: Stress variables

Variable name	Variable location type	Notes
AXIAL_STRESS	Element field	18
DISTORTIONAL_STRESS	Element field	
EFFECTIVE_STRESS	Element field	1,3
EQUIV_INTERNAL_AXIAL_PRESSURE	Element field	
EQUIV_INTERNAL_HOOP_PRESSURE	Element field	
FE_EFFECTIVE_STRESS	Element field	2,3
FE_MAX_SHEAR_STRESS	Element field	2
FE_PRESSURE	Element field	2,4
FE_SIGMA-P1	Element field	2,5
FE_SIGMA-P1_ANGLE	Element field	2,5
FE_SIGMA-P1_DIRECTION-(XYZ)	Element field	2,5
FE_SIGMA-P2	Element field	2,5
FE_SIGMA-P2_ANGLE	Element field	2,5
FE_SIGMA-P2_DIRECTION-(XYZ)	Element field	2,5
FE_SIGMA-P3	Element field	2,5
FE_SIGMA-P3_DIRECTION-(XYZ)	Element field	2,5
FLUID_REFERENCE_PRESSURE	Element field	
GASKET_PRESSURE	Element field	
GASKET_STRESS-(ABC)	Element field	17
GASKET_YIELD_STRESS	Element field	
GENERAL_ELEMENT_STRESS	Element field	
GRAVITY_IN-SITU_PRESSURE	Element field	
HILL_EFFECTIVE_STRESS	Element field	
MAX_SHEAR_STRESS	Element field	1,7
MEAN_STRESS	Element field	
NODAL_PRESSURE	Node field	4
PORE_PRESSURE	Node field	15
PRESSURE	Element field	1,4
PRESSURE_GRADIENT_MAGNITUDE	Element field	16
SIGMA-NORM2	Element field	1,8
SIGMA-(P1,P2,P3)	Element field	1,6
SIGMA-(P1,P2,P3)_DIRECTION-(XYZ)	Element field	1,14
STRESS-(123)	Element field	9
STRESS-(ABC)	Element field	10
STRESS-(IJK)	Element field	11
STRESS-(RST)	Element field	12

STRESS- (XYZ)	Element field	13
STRESS_THRU_THICKNESS	Element field	
YIELD_STRESS	Element field	

1) These variables are computed by the AUI from data output by the solution program.

2) Variables that begin with FE_ denote quantities that are output directly from the solution program. For example, FE_EFFECTIVE_STRESS is the effective stress as output from ADINA for certain elements/material models.

These variables should be distinguished from variables defined in the AUI as predefined resultants (see Note 1).

3) Variable FE_EFFECTIVE_STRESS is the effective stress as output by ADINA. Usually FE_EFFECTIVE_STRESS and EFFECTIVE_STRESS are numerically identical to each other, when they are evaluated at the element integration points. One case in which FE_EFFECTIVE_STRESS and EFFECTIVE_STRESS are not numerically identical is when ADINA uses a large displacement/small strain formulation. In this case, ADINA uses 2nd Piola-Kirchhoff stresses internally and variable FE_EFFECTIVE_STRESS is a 2nd Piola-Kirchhoff stress; but since variables STRESS-XX, etc are Cauchy stresses, EFFECTIVE_STRESS is also a Cauchy stress.

The AUI interpolates (and possibly smoothes) EFFECTIVE_STRESS from the integration points differently than it interpolates FE_EFFECTIVE_STRESS. The AUI interpolates EFFECTIVE_STRESS by interpolating the stress components (possibly smoothing them as well) individually; then, after interpolation, the AUI computes EFFECTIVE_STRESS using the formula

$$\sigma_e = \sqrt{1/2 \left((\tau_{xx} - \tau_{yy})^2 + (\tau_{xx} - \tau_{zz})^2 + (\tau_{yy} - \tau_{zz})^2 + 6(\tau_{xy}^2 + \tau_{xz}^2 + \tau_{yz}^2) \right)}$$

However, the AUI interpolates and smoothes FE_EFFECTIVE_STRESS using the same algorithms that it uses for any other stress component.

4) The comments given in Note 3 apply to the variables PRESSURE (computed as

$$p = -\frac{\tau_{xx} + \tau_{yy} + \tau_{zz}}{3} \text{ by the AUI) and FE_PRESSURE. The variable}$$

NODAL_PRESSURE is output by ADINA-F (since pressure is calculated at the nodes

in ADINA-F), and, when postprocessing ADINA-F results, you can use `NODAL_PRESSURE` to plot or list the nodal pressures.

5) These are the principal stresses and their directions, as output by the ADINA program. Note that the concrete and curve description models, these are the true principal stresses only before cracking has occurred. After cracking, the directions are fixed corresponding to the crack directions and these variables are no longer principal stresses.

For the data fitted concrete material model, `FE_SIGMA-P1_ANGLE` and `FE_SIGMA-P2_ANGLE` for 2-D solid elements, and `FE_SIGMA-P1_DIRECTION-(XYZ)`, `FE_SIGMA-P2_DIRECTION-(XYZ)` and `FE_SIGMA-P3_DIRECTION-(XYZ)` for 3-D solid elements define the normal directions of the crack planes; `FE_SIGMA-P1`, `FE_SIGMA-P2` and `FE_SIGMA-P3` are the stress values at the time of crack formation along the normal directions of the crack planes.

For 2-D solid elements, `FE_SIGMA-P3` is always out of the plane of the element and `FE_SIGMA-P1`, `FE_SIGMA-P2` are always in the plane of the element ($FE_SIGMA-P1 \geq FE_SIGMA-P2$) only for the concrete and curve description material models.

6) These are the principal stresses as computed by the AUI from the stress components ($SIGMA-P1 \geq SIGMA-P2 \geq SIGMA-P3$).

7) This is the maximum shear stress, defined as one-half the difference between the maximum and minimum principal stress, as computed by the AUI from the stress components.

8) This is the 2-norm of the stress tensor (the larger of the absolute values of the maximum and minimum principal stress) as computed by the AUI from the stress components.

9) These are the stresses transformed by the AUI into a user-specified coordinate system. More information is given at the end of this section.

Note that these variables have nothing to do with the initial stresses entered in the initial strain directions.

10) These are the stress components in the axes of the material coordinate system, for orthotropic materials. Note that the AUI cannot transform stresses output in other coordinate systems to the material coordinate system.

11) These are equal to the global stress components for those elements in which the

stresses are output in the global coordinate system, equal to the isoparametric coordinate system stress components for those elements in which the stresses are output in the isoparametric coordinate system, and equal to the material coordinate system stress components for those elements in which the stresses are output in the material coordinate system. These variables are mainly used for constructing resultants that are invariants (that is, do not depend on the coordinate system direction). For example, resultant variable `EFFECTIVE_STRESS` is defined in terms of `STRESS-II`, `STRESS-JJ`, etc.

12) These are the stress components in the axes of the isoparametric coordinate system. For shell elements, component directions R and S refer to directions \bar{r} and \bar{s} (see Section 2.7.1 and Figure 2.7-6). Note that the AUI cannot transform stresses output in other coordinate systems to the isoparametric coordinate system.

13) These are the stress components in the global coordinate system. Note that the AUI cannot transform stresses output by ADINA in other coordinate systems to the global coordinate system.

14) These are the directions of the principal stresses `SIGMA-P1`, `SIGMA-P2`, `SIGMA-P3` (see note 6).

15) This is the value of the pore pressure, used in the porous media formulation.

16) `PRESSURE_GRADIENT_MAGNITUDE` is implemented for the ADINA-F 3- and 4-node 2D elements, and for the 4- and 8-node 3D elements. The purpose of `PRESSURE_GRADIENT_MAGNITUDE` is to provide an ADINA-F variable that should be smooth in the mathematical solution but that may not be smooth in the finite element solution. A band plot of `PRESSURE_GRADIENT_MAGNITUDE` will show jumps between elements.

17) In 2-D analysis, the BB, AB and CC components are defined; in 3-D analysis, the BB, CC, AB, AC, BC components are defined.

18) In band plots, `AXIAL_STRESS` can be used instead of `STRESS-RR`.

Table 13.1-8: Strain variables

Variable name	Variable location type	Notes
<code>ACCUM_EFF_CREEP_STRAIN</code>	Element field	11
<code>ACCUM_EFF_PLASTIC_STRAIN</code>	Element field	5

ACCUM_EFF_TRANSF_STRAIN	Element field	12
ACCUM_EFF_VISCOPLASTIC_STRAIN	Element field	
ACCUM_PLASTIC_AXIAL_STRAIN	Element field	
ACCUM_PLASTIC_CURVATURE- (ST)	Section field	
ACCUM_PLASTIC_TWIST	Section field	
AUSTENITE_FRACTION	Element field	12
AXIAL_STRAIN	Section field	
CREEP_STRAIN- (ABC)	Element field	1
CREEP_STRAIN- (RST)	Element field	1
CREEP_STRAIN- (XYZ)	Element field	1
CREEP_STRAIN_THRU_THICKNESS	Element field	
CURVATURE- (RB, SB, RBSB)	Section field	
CURVATURE- (ST)	Section field	
CURVATURE- (XL, YL, XLYL)	Element field	
DEFORMATION_GRADIENT- (XYZ)	Element field	6
DETWINNED_MARTENSITE_FRACTION	Element field	12
EFF_VISCOPLASTIC_STRAIN_RATE	Element field	
EFFECTIVE_CREEP_STRAIN	Element field	
FE_STRAIN-P1	Element field	14
FE_STRAIN-P2	Element field	14
FE_STRAIN-P3	Element field	14
GASKET_CLOSURE_STRAIN	Element field	
GASKET_PLASTIC_CLOSURE_STRAIN	Element field	
GASKET_STRAIN- (ABC)	Element field	9
GASKET_THERMAL_STRAIN	Element field	
GASKET_THERMAL_STRAIN- (ABC)	Element field	9
IRRADIATION_STRAIN	Element field	
LOGSTRAIN_P1	Element field	10
LOGSTRAIN_P2	Element field	10
LOGSTRAIN_P3	Element field	10
LSTRETCH- (XYZ)	Element field	7
MEMBRANE_STRAIN- (RB, SB, RBSB)	Section field	
MEMBRANE_STRAIN- (XL, YL, XLYL)	Element field	
NEUTRAL_AXIS_POSITION- (RB, SB)	Section field	
PLASTIC_AXIAL_STRAIN	Section field	
PLASTIC_CURVATURE- (ST)	Section field	
PLASTIC_STRAIN- (ABC)	Element field	1
PLASTIC_STRAIN- (RST)	Element field	1
PLASTIC_STRAIN- (XYZ)	Element field	1

PLASTIC_STRAIN_THRU_THICKNESS	Element field	
PLASTIC_TWIST	Section field	
RSTRETCH- (XYZ)	Element field	8
SHEAR_STRAIN- (RB, SB)	Section field	
STRAIN- (123)	Element field	3
STRAIN- (ABC)	Element field	1,4
STRAIN- (IJK)	Element field	1,4
STRAIN- (RST)	Element field	1,4
STRAIN- (XYZ)	Element field	1,4
STRAIN- (P1, P2, P3)	Element field	1,13
STRAIN_THRU_THICKNESS	Element field	
STRETCH- (XYZ)	Element field	2
THERMAL_STRAIN	Element field	
THERMAL_STRAIN- (ABC)	Element field	1
THERMAL_STRAIN- (RST)	Element field	1
THERMAL_STRAIN- (XYZ)	Element field	1
THERMAL_STRAIN_THRU_THICKNESS	Element field	
TRANSFORMATION_STRAIN- (RST)	Element field	12
TRANSFORMATION_STRAIN- (XYZ)	Element field	12
TWINNED_MARTENSITE_FRACTION	Element field	12
TWIST	Section field	
VISCOPLASTIC_STRAIN- (XYZ)	Element field	1
VOLUMETRIC_STRAIN	Element field	

1) The off-diagonal strain terms are engineering strains; i.e. twice the tensorial strains.

2) These are the Cartesian components of the left stretch tensor \mathbf{V} , as output by ADINA. See Section 3.1, equation 3.1-5 for the definition and meaning of \mathbf{V} .

The principal stretches can be computed in the AUI by defining a resultant using functions SIGP1, SIGP2 and SIGP3. For example, you can define, for 3-D problems,

```
MAX_STRETCH = SIGP1 (<STRETCH-XX>, <STRETCH-YY>,
                    <STRETCH-ZZ>, <STRETCH-XY>,
                    <STRETCH-XZ>, <STRETCH-YZ>)
```

The left stretch tensor is also used in the AUI for plotting the stretches as element vectors.

3) These are the strain components transformed by the AUI into a user-specified

coordinate system. See the end of this section for more information.

Note that these strains have nothing to do with the initial strains entered in the initial strain directions.

4) The same comments as are given for the stress variables (notes 10, 12, 13 of Table 13.1-7) apply to the strain variables.

5) The definition of the accumulated effective plastic strain is given in Section 3.4.1.

6) These are the components of the deformation gradient tensor ${}^t_0\mathbf{X}$, as output from ADINA. Note that there are 9 components, since the deformation gradient tensor is in general not symmetric.

7) These are the Cartesian components of the left stretch tensor \mathbf{V} , as computed by the AUI from the deformation gradients output by ADINA. See Section 3.1, equation 3.1-5 and Figure 3.1-2 for the definition and meaning of \mathbf{V} .

The principal stretches can be computed in the AUI by defining a resultant using functions SIGP1, SIGP2 and SIGP3. For example, you can define, for 3-D problems,

```
MAX_STRETCH = SIGP1 (<LSTRETCH-XX>, <LSTRETCH-YY>,  
                    <LSTRETCH-ZZ>, <LSTRETCH-XY>,  
                    <LSTRETCH-XZ>, <LSTRETCH-YZ>)
```

The left stretch tensor is also used in the AUI for plotting the stretches as element vectors.

8) These are the Cartesian components of the right stretch tensor \mathbf{U} , as computed by the AUI from the deformation gradients output by ADINA. See Section 3.1, equation 3.1-1 for the definition and meaning of \mathbf{U} .

The principal stretches can be computed in the AUI by defining a resultant using functions SIGP1, SIGP2 and SIGP3. For example, you can define, for 3-D problems,

```
MAX_STRETCH = SIGP1 (<RSTRETCH-XX>, <RSTRETCH-YY>,  
                    <RSTRETCH-ZZ>, <RSTRETCH-XY>,  
                    <RSTRETCH-XZ>, <RSTRETCH-YZ>)
```

9) In 2-D analysis, the BB, AB and CC components are defined; in 3-D analysis, the BB, CC, AB, AC, BC components are defined.

10) These are the maximum, intermediate and minimum principal logarithmic strains. They are available in large strain analysis.

- 11) ACCUM_EFF_CREEP_STRAIN is defined in Section 3.6.3.
- 12) These variables are used with the SMA material defined in Section 3.14.
- 13) These are the maximum intermediate and minimum principal strains, as computed from variables STRAIN-II, STRAIN-IJ, etc.
- 14) These variables apply to the data fitted concrete material model. They are the strain values at the time of crack formation along the normal directions of the crack planes.

Table 13.1-9: Force variables

Variable name	Variable location type	Notes
AVERAGE_BENDING_MOMENT	Section field	1,6
AVERAGE_MEMBRANE_FORCE	Section field	2,6
AVERAGE_SHEAR_FORCE	Section field	6
AXIAL_FORCE	Section field	
BENDING_MOMENT- (RB, SB, RBSB)	Section field	3
BENDING_MOMENT- (ST)	Section field	
BENDING_MOMENT- (XL, YL, XLYL)	Element field	4
CONTACT_FORCE- (123)	Node discrete	15
CONTACT_FORCE- (ABC)	Node discrete	
CONTACT_FORCE- (XYZ)	Node discrete	
DAMPING_FORCE	Element field	
EFFECTIVE_BENDING_MOMENT	Section field	
EFFECTIVE_MEMBRANE_FORCE	Section field	
ELASTIC_FORCE	Element field	
FORCE- (ABC)	Element field	
FORCE- (RST)	Element field	
FORCE- (XYZ)	Element field	
FORCE_RST_MAGNITUDE	Element field	11
MAX_PRINCIPAL_BENDING_MOMENT	Section field	6, 7
MAX_PRINCIPAL_MEMBRANE_FORCE	Section field	6, 8
MEMBRANE_FORCE- (RB, SB, RBSB)	Section field	3
MEMBRANE_FORCE- (XL, YL, XLYL)	Element field	4
MIN_PRINCIPAL_BENDING_MOMENT	Section field	6, 9
MIN_PRINCIPAL_MEMBRANE_FORCE	Section field	6, 10
MOMENT- (ABC)	Element field	

MOMENT- (RST)	Element field	
MOMENT- (XYZ)	Element field	
MOMENT_RST_MAGNITUDE	Element field	11
NODAL_FORCE- (RST)	El local node	5
NODAL_FORCE- (XYZ)	El local node	5, 14
NODAL_FORCE_RST_MAGNITUDE	El local node	11
NODAL_MOMENT- (RST)	El local node	5
NODAL_MOMENT- (XYZ)	El local node	5, 14
NODAL_MOMENT_RST_MAGNITUDE	El local node	11
RESTRAINING_FORCE- (XYZ)	Node discrete	12
SHEAR_FORCE- (RB, SB)	Section field	3
SOLITARY_CONTACT_FORCE- (ABC)	Node discrete	
SOLITARY_CONTACT_FORCE- (XYZ)	Node discrete	
TORSIONAL_MOMENT	Section field	
UPLIFTING_FORCE- (XYZ)	Node discrete	13
YIELD_AXIAL_FORCE	Section field	
YIELD_BENDING_MOMENT- (ST)	Section field	
YIELD_TORSIONAL_MOMENT	Section field	

1) This is evaluated by the AUI as $0.5(M_{\bar{r}\bar{r}} - M_{\bar{s}\bar{s}})$, which is an invariant (does not depend on the directions of \bar{r} and \bar{s}).

2) This is evaluated by the AUI as $0.5(R_{\bar{r}\bar{r}} - R_{\bar{s}\bar{s}})$, which is an invariant (does not depend on the directions of \bar{r} and \bar{s}).

3) RB denotes the \bar{r} direction, SB denotes the \bar{s} direction.

4) XL denotes the element local x direction, YL denotes the element local y direction.

5) These are the consistent nodal point forces corresponding to the stresses within the element. The AUI does not transform these from the coordinate system in which they are output by ADINA.

6) These are computed by the AUI from the data output by ADINA.

7) This is evaluated by the AUI as the maximum principal value of $\begin{bmatrix} -M_{\bar{s}\bar{s}} & M_{\bar{r}\bar{s}} \\ M_{\bar{r}\bar{s}} & M_{\bar{r}\bar{r}} \end{bmatrix}$.

8) This is evaluated by the AUI as the maximum principal value of $\begin{bmatrix} R_{\overline{F}\overline{F}} & R_{\overline{F}\overline{S}} \\ R_{\overline{F}\overline{S}} & R_{\overline{S}\overline{S}} \end{bmatrix}$.

9) This is evaluated by the AUI as the minimum principal value of $\begin{bmatrix} -M_{\overline{S}\overline{S}} & M_{\overline{F}\overline{S}} \\ M_{\overline{F}\overline{S}} & M_{\overline{F}\overline{F}} \end{bmatrix}$.

10) This is evaluated by the AUI as the minimum principal value of $\begin{bmatrix} R_{\overline{F}\overline{F}} & R_{\overline{F}\overline{S}} \\ R_{\overline{F}\overline{S}} & R_{\overline{S}\overline{S}} \end{bmatrix}$.

11) Each of these variables is the square root of the sum of the squares of the components, as computed by the AUI.

12) `RESTRAINING_FORCE-(XYZ)` is the restraining forces on the nodes of the drawbead segments.

13) `UPLIFTING_FORCE-(XYZ)` is the uplifting forces on the nodes of the drawbead segments.

14) For 2D, 3D, plate, shell and spring elements, these variables are output in the coordinate system of the node to which the element is attached (global system or skew system). For example, if a 3D element is attached to nodes with skew systems, variable `NODAL_FORCE-X` is the nodal force in the skew A direction.

15) These are the components of the contact forces, transformed by the AUI into a user-specified coordinate system. More information is given at the end of this section.

Table 13.1-10: Traction variables

Variable name	Variable location type	Notes
<code>NODAL_NORMAL_TRACTION</code>	Node field	6, 7
<code>NODAL_NORMAL_TRACTION-(123)</code>	Node field	6,10
<code>NODAL_NORMAL_TRACTION-(XYZ)</code>	Node field	6
<code>NODAL_TANGENTIAL_TRACTION</code>	Node field	8, 9
<code>NODAL_TANGENTIAL_TRACTION-(123)</code>	Node field	8,10
<code>NODAL_TANGENTIAL_TRACTION-(XYZ)</code>	Node field	8
<code>NORMAL_TRACTION</code>	Contact	1

NORMAL_TRACTION- (XYZ)	Contact	
RESTRAINING_TRACTION- (XYZ)	Drawbead segment	4
SURFACE_TRACTION- (XYZ)	Mesh surface	2
TANGENTIAL_TRACTION	Contact	3
TANGENTIAL_TRACTION- (XYZ)	Contact	

1) This is the magnitude of the normal traction vector.

2) These are computed by the AUI as predefined resultants using the formulas

$$\begin{aligned} \text{SURFACE_TRACTION-X} = & \langle \text{STRESS-XX} \rangle * \langle \text{SURFACE_NORMAL-X} \rangle + \\ & \langle \text{STRESS-XY} \rangle * \langle \text{SURFACE_NORMAL-Y} \rangle + \\ & \langle \text{STRESS-XZ} \rangle * \langle \text{SURFACE_NORMAL-Z} \rangle \end{aligned}$$

$$\begin{aligned} \text{SURFACE_TRACTION-Y} = & \langle \text{STRESS-XY} \rangle * \langle \text{SURFACE_NORMAL-X} \rangle + \\ & \langle \text{STRESS-YY} \rangle * \langle \text{SURFACE_NORMAL-Y} \rangle + \\ & \langle \text{STRESS-YZ} \rangle * \langle \text{SURFACE_NORMAL-Z} \rangle \end{aligned}$$

$$\begin{aligned} \text{SURFACE_TRACTION-Z} = & \langle \text{STRESS-XZ} \rangle * \langle \text{SURFACE_NORMAL-X} \rangle + \\ & \langle \text{STRESS-YZ} \rangle * \langle \text{SURFACE_NORMAL-Y} \rangle + \\ & \langle \text{STRESS-ZZ} \rangle * \langle \text{SURFACE_NORMAL-Z} \rangle \end{aligned}$$

See Table 13.1-17 for the definitions of the surface normal variables.

3) This is the magnitude of the tangential traction vector.

4) RESTRAINING_TRACTION- (XYZ) is the restraining tractions on the drawbead segments.

5) UPLIFTING_TRACTION- (XYZ) is the uplifting tractions on the drawbead segments.

6) These variables are output when the “new” contact segments are used.

7) NODAL_NORMAL_TRACTION is the magnitude of the normal traction.

8) These variables are output when the “new” contact segments are used in frictional contact.

9) NODAL_TANGENTIAL_TRACTION is the magnitude of the tangential traction.

10) These are the components of the nodal normal and tangential tractions, transformed by the AUI into a user-specified coordinate system. More information is given at the end of this section.

Table 13.1-11: Reaction variables

Variable name	Variable location type	Notes
(123)-MOMENT_REACTION	Node discrete	1
(ABC)-MOMENT_REACTION	Node discrete	2
(V1,V2)-MOMENT_REACTION	Node discrete	2
(123)-MOMENT_REACTION	Node discrete	1
(XYZ)-MOMENT_REACTION	Node discrete	2
(123)-MOMENTUM_FLUX_ELEDGE	Element face	1
(XYZ)-MOMENTUM_FLUX_ELEDGE	Element face	4
(123)-MOMENTUM_FLUX_ELFACE	Element face	1
(XYZ)-MOMENTUM_FLUX_ELFACE	Element face	5
(ABC)-REACTION	Node discrete	2
(XYZ)-REACTION	Node discrete	2
(123)-REACTION_ELEDGE	Element face	1
(XYZ)-REACTION_ELEDGE	Element face	4
(123)-REACTION_ELFACE	Element face	1
(XYZ)-REACTION_ELFACE	Element face	5
REACTION_MAGNITUDE	Node discrete	3

1) These are the components of the reactions, moment reactions, etc., transformed by the AUI into a user-specified coordinate system. More information is given at the end of this section.

2) Reactions are only output for those nodes with fixities. Reactions that are numerically zero might not be output by ADINA.

3) REACTION_MAGNITUDE is the square root of the sum of the squares of the reaction components, as computed by the AUI.

4) These are the components of momentum flux output on ADINA-F element edges (2D analysis). On boundaries with no flow, momentum flux is equal to reaction.

5) These are the components of momentum flux output on ADINA-F element faces (3D analysis). On boundaries with no flow, momentum flux is equal to reaction.

Table 13.1-12: Flux variables

Variable name	Variable location type	Notes
---------------	------------------------	-------

AXIAL_HEAT_FLUX	Element field	
HEAT_FLUX_SURFACE	Mesh surface	1
HEAT_FLUX-(123)	Element field	7
HEAT_FLUX-(XYZ)	Element field	
HEAT_FLUX_MAGNITUDE	Element field	6
KINETIC_ENERGY_FLUX_SURFACE	Mesh surface	2
MASS_FLUX_ELEDGE	Element face	8
MASS_FLUX_ELFACE	Element face	9
MASS_FLUX_SURFACE	Mesh surface	3
MOMENTUM_FLUX-(XYZ)_SURFACE	Mesh surface	4
RADIATION_HEAT_FLUX-(XYZ)	Radiation field	
SEEPAGE_FLUX-(123)	Element field	7
SEEPAGE_FLUX-(XYZ)	Element field	
SEEPAGE_FLUX_MAGNITUDE	Element field	6
VOLUME_FLUX_SURFACE	Mesh surface	5

1) This is computed by the AUI as a predefined resultant using the formula

$$\text{HEAT_FLUX_SURFACE} = \langle \text{HEAT_FLUX-X} \rangle * \langle \text{SURFACE_NORMAL-X} \rangle + \\ \langle \text{HEAT_FLUX-Y} \rangle * \langle \text{SURFACE_NORMAL-Y} \rangle + \\ \langle \text{HEAT_FLUX-Z} \rangle * \langle \text{SURFACE_NORMAL-Z} \rangle$$

See Table 13.1-17 for the definitions of the surface normal variables.

2) This is computed by the AUI as a predefined resultant using the formula

$$\text{KINETIC_ENERGY_FLUX_SURFACE} = \\ \text{KINETIC_ENERGY_DENSITY} * \text{VOLUME_FLUX_SURFACE}$$

3) This is computed by the AUI as a predefined resultant using the formula

$$\text{MASS_FLUX_SURFACE} = \text{ELEMENT_DENSITY} * \\ (\langle \text{X-VELOCITY} \rangle * \langle \text{SURFACE_NORMAL-X} \rangle + \\ \langle \text{Y-VELOCITY} \rangle * \langle \text{SURFACE_NORMAL-Y} \rangle + \\ \langle \text{Z-VELOCITY} \rangle * \langle \text{SURFACE_NORMAL-Z} \rangle)$$

See Table 13.1-17 for the definitions of the surface normal variables. You must define variable ELEMENT_DENSITY before using variable MASS_FLUX_SURFACE.

4) These are computed by the AUI as predefined resultants using the formulas

$$\text{MOMENTUM_FLUX-X_SURFACE} = \langle \text{X-VELOCITY} \rangle * \langle \text{MASS FLUX SURFACE} \rangle$$
$$\text{MOMENTUM_FLUX-Y_SURFACE} = \langle \text{Y-VELOCITY} \rangle * \langle \text{MASS FLUX SURFACE} \rangle$$
$$\text{MOMENTUM_FLUX-Z_SURFACE} = \langle \text{Z-VELOCITY} \rangle * \langle \text{MASS_FLUX_SURFACE} \rangle$$

You must define variable `ELEMENT_DENSITY` before using these variables. See Table 13.1-17 for the definitions of the surface normal variables.

5) This is computed by the AUI as a predefined resultant using the formula

$$\text{VOLUME_FLUX_SURFACE} = \langle \text{X-VELOCITY} \rangle * \langle \text{SURFACE_NORMAL-X} \rangle + \langle \text{Y-VELOCITY} \rangle * \langle \text{SURFACE_NORMAL-Y} \rangle + \langle \text{Z-VELOCITY} \rangle * \langle \text{SURFACE_NORMAL-Z} \rangle$$

See Table 13.1-18 for the definitions of the surface normal variables.

6) Each of these variables is the square root of the sum of the squares of their components, as computed by the AUI.

7) These are the components of the fluxes, transformed by the AUI into a user-specified coordinate system. More information is given at the end of this section.

8) This is the mass flux output at ADINA-F element edges (2D analysis). The mass flux is positive when the mass flux is in the same direction as the element edge inwards normal.

9) This is the mass flux output at ADINA-F element faces (3D analysis). The mass flux is positive when the mass flux is in the same direction as the element face inwards normal.

Table 13.1-13: Electric field variables

Variable name	Variable location type	Notes
ELECTRIC_POTENTIAL	Node field	
CURRENT_DENSITY-(123)	Element field	2
CURRENT_DENSITY-(XYZ)	Element field	
CURRENT_DENSITY_MAGNITUDE	Element field	1

HEAT DISSIPATION	Element field
HEAT DISSIPATION RATE	Element field
TOTAL HEAT DISSIPATION	Loc-independent
TOTAL HEAT DISSIPATION RATE	Loc-independent

1) CURRENT_DENSITY_MAGNITUDE is the square root of the sum of the squares of the current density components, as computed by the AUI.

2) These are the components of the current density, transformed by the AUI into a user-specified coordinate system. More information is given at the end of this section.

Table 13.1-14: Fracture variables

Variable name	Variable location type	Notes
CRACKED_SURFACE	Virtual shift	1
CRACK_BOUNDARY_FLAG	Node discrete	2
CRACK_LENGTH	Loc-independent	
DYNAMIC_CORRECTION	Virtual shift	3
J-PARAMETER_1	Virtual shift	3,4
J-PARAMETER_2	Virtual shift	3,5
J-PARAMETER_3	Virtual shift	3,6
LINE_CONTOUR_NUMBER	Line contour	7
LINE_J-INTEGRAL_AVERAGE	Line contour	7,8
LINE_J-INTEGRAL_DIFFERENCE	Line contour	7
PRESSURE_CORRECTION	Virtual shift	3
TEMPERATURE_CORRECTION	Virtual shift	3
VIRTUAL_CRACK_SURFACE	Virtual shift	3,9
VIRTUAL_SHIFT_NUMBER	Virtual shift	3
VIRTUAL_SHIFT*_PART_FACTOR	Node field	3,10
(XYZ)-VIRTUAL_SHIFT	Virtual shift	3,11

1) This variable is used to plot the cracked surface in 3-D crack propagation analysis (ADINA fracture mechanics analysis). It is equal to 1.0 if the node is on the crack front or has been released; otherwise it is not defined. This variable can be plotted in a band plot to shade the area that has cracked during the analysis.

2) This variable indicates the current boundary condition applied to a node on a crack propagation surface (used in ADINA fracture mechanics analysis). It has the following values: 0=node is released, 1=node is on the crack front and is fixed,

2=node is on the crack front and is constrained, 11=node is unreleased and is fixed, 12=node is unreleased and is constrained.

The crack boundary flag is not defined on nodes that are not on the crack propagation surface.

3) These variables are used in ADINA fracture mechanics analysis, virtual crack extension method, see Chapter 10.

4) This value of the J-integral is obsolete.

5) This value of the J-integral does not include temperature, pressure or dynamic corrections.

6) This value of the J-integral includes temperature, pressure and dynamic corrections.

7) These variables are used in ADINA fracture mechanics analysis, line contour method.

8) This is the average of the J-integrals for the outer and inner paths within elements that use 2x2 Gauss integration.

9) This is the crack surface area.

10) This variable gives, for each node, the amount of participation in the virtual shift, from "undefined" (node is not included in virtual shift) to 0.0 (node is included, but does not shift) to 1.0 (node is included and shifts completely). It can be used in a band plot to show the virtual shift.

11) These are the components of the virtual shift vector.

Table 13.1-15: Failure criterion variables

Variable name	Variable location type	Notes
CAM-CLAY_FLAG	Element field	14
CAP_LOCATION	Element field	
CRACK_FLAG	Element field	1
DF-CONCRETE_FLAG	Element field	17
DRUCKER-PRAGER_FLAG	Element field	16
EFFECTIVE_STRESS_RATIO	Element field	

FAILURE_CRITERION	Element field	2,3
FAILURE_CRITERION_COMP-FIBER	Element field	2,3
FAILURE_CRITERION_COMP-MATRIX	Element field	2,3
FAILURE_CRITERION_SURFACE- (1234)	Element field	2,3
FAILURE_CRITERION_TENS-FIBER	Element field	2,3
FAILURE_CRITERION_TENS-MATRIX	Element field	2,3
FAILURE_FLAG	Element field	2,4
FAILURE_FLAG_COMP-FIBER	Element field	2,5
FAILURE_FLAG_COMP-MATRIX	Element field	2,5
FAILURE_FLAG_COMPRESSION- (ABC)	Element field	2,5
FAILURE_FLAG_SHEAR- (AB, AC, BC)	Element field	2,5
FAILURE_FLAG_SURFACE- (1234)	Element field	2,5
FAILURE_FLAG_TENS-FIBER	Element field	2,5
FAILURE_FLAG_TENS-MATRIX	Element field	2,5
FAILURE_FLAG_TENSION- (ABC)	Element field	2,5
FAILURE_SURFACE_FUNCTION	Element field	10
GASKET_DEFORMATION_MODE	Element field	11
GASKET_STATUS	Element field	12
MOHR-COULOMB_FLAG	Element field	15
NUMBER_OF_CRACKS	Element field	6
PLASTIC_FLAG	Element field	7
PLASTIC_FLAG_2	Element field	8
PLASTIC_FLAG_AXIAL	Section field	
PLASTIC_FLAG_BENDING- (ST)	Section field	
PLASTIC_FLAG_TORSION	Section field	
SMA_FLAG	Element field	13
SPECIFIC_VOLUME	Element field	
VOID_RATIO	Element field	
VOID_VOLUME_FRACTION	Element field	
WRINKLE_FLAG	Element field	9
YIELD_FUNCTION	Element field	10
YIELD_SURFACE_DIAMETER_P	Element field	

1) This is a three-digit integer (P1)(P2)(P3) interpreted as follows. Each digit represents a principal stress coordinate direction. If the digit equals 0 (zero), there is no crack; if the digit equals 1, there is an open crack; if the digit equals 2, there is a closed crack. If the crack flag equals 333, the material is crushed.

The related variable NUMBER_OF_CRACKS returns the number of cracks at the point, as computed from this variable.

2) The failure variables are used in analyses employing the anisotropic failure criteria within shell element groups, see Section 2.7.8.

3) Each failure criterion variable stores the value of the failure criterion used in the analysis.

4) The failure flag has the following values: 0=material at the point considered has not failed, 1=material at the point considered has failed.

5) All of the failure flags with a qualifier have the following values: 0=material at the point considered has not failed in that mode, 1=material at the point considered has failed in that mode.

6) This variable returns the number of cracks at the point, as computed by the AUI from the variable `CRACK_FLAG`. If the material is crushed, the value of `CRACK_FLAG` is 4.

7) The plastic flag has the following values: 1=elastic conditions, 2=plastic conditions.

For the Mohr-Coulomb material, the plastic flag can have the following values: 1=elastic conditions; 2=Mohr-Coulomb plastic deformation; 3=plastic deformation at the yield surface vertices (Drucker-Prager yielding); 4=tension cutoff. Note that, for the Mohr-Coulomb material, variable `MOHR-COULOMB_FLAG` should be used instead of `PLASTIC_FLAG`. `MOHR-COULOMB_FLAG` has the same numerical values as `PLASTIC_FLAG`, and the AUI knows the allowed values of `MOHR-COULOMB_FLAG`.

8) This variable is used for the Drucker-Prager material model. It has the following values: 1=elastic conditions, 2=Drucker-Prager yielding, 3=cap yielding, 4=vertex yielding, 5=tension cutoff yielding. Note that, for the Drucker-Prager material, variable `DRUCKER-PRAGER_FLAG` should be used instead of `PLASTIC_FLAG_2`. `DRUCKER-PRAGER_FLAG` has the same numerical values as `PLASTIC_FLAG_2`, and the AUI knows the allowed values of `DRUCKER-PRAGER_FLAG`.

9) This variable has the following values: 0=no wrinkling, 1=wrinkling in one direction, 2=wrinkling in two directions.

10) This variable is the value of the yield function of failure surface function.

11) `GASKET_DEFORMATION_MODE` has the following values: 1=elastic, 2=plastic, 3=rupture.

12) GASKET_STATUS has the following values: 1=open, 2=closed, 3=sealed, 4=leaked, 5=crushed.

13) SMA_FLAG is used with the SMA material. It has the following values: 1=elastic state; 2=martensite transformation; 3=austenite transformation.

14) CAM-CLAY_FLAG is used with the Cam-clay material. It has the following values: 1=elastic state, 2=hardening, 3=softening, 4=critical.

15) MOHR-COULOMB_FLAG is used with the Mohr-Coulomb material. It has the following values: 1=elastic state, 2=Mohr-Coulomb yielding, 3=vertex yielding (Drucker-Prager yielding), 4=tension cutoff yielding.

16) DRUCKER-PRAGER_FLAG is used with the Drucker-Prager material. It has the following values: 1=elastic state, 2=Drucker-Prager yielding, 3=cap yielding, 4=vertex yielding, 5=tension cutoff yielding.

17) DF-CONCRETE_FLAG is used with the data fitted concrete material model. It has the following values: 1 = elastic unloading; 2 = elastic reloading; 3 = nonlinear loading; 4 = one-crack; 5 = two-crack; 6 = fully cracked; and 7 = crushed.

Table 13.1-16: Eigenvector variables

Variable name	Variable location type	Notes
(123)-EIGENVECTOR	Node field	1
(ABC)-EIGENVECTOR	Node field	
(XYZ)-EIGENVECTOR	Node field	
(123)-EIGENVECTOR_ROTATION	Node field	1
(ABC)-EIGENVECTOR_ROTATION	Node field	
(V1,V2)-EIGENVECTOR_ROTATION	Node field	
(XYZ)-EIGENVECTOR_ROTATION	Node field	2
EIGENVECTOR_MAGNITUDE	Node field	
FLUID_P0_EIGENVECTOR	Node field	
FLUID_POTENTIAL_EIGENVECTOR	Node field	
OVALIZATION_EIGENVECTOR-(123456)	Node field	
PORE_PRESSURE_EIGENVECTOR	Node field	
THERMAL_EIGENVECTOR	Node field	

1) These are the components of the eigenvectors transformed by the AUI into a user-

specified coordinate system. More information is given at the end of this section.

2) `EIGENVECTOR_MAGNITUDE` is the square root of the sum of the squares of the eigenvector components, as computed by the AUI.

Table 13.1-17: Prescribed load variables

Variable name	Variable location type	Notes
<code>(ABC)-PRESCRIBED_ACCELERATION</code>	Node field	
<code>(XYZ)-PRESCRIBED_ACCELERATION</code>	Node field	
<code>(ABC)-PRESCRIBED_ANG_ACCELERATION</code>	Node field	
<code>(V1,V2)-PRESCRIBED_ANG_ACCELERATION</code>	Node field	
<code>(XYZ)-PRESCRIBED_ANG_ACCELERATION</code>	Node field	
<code>PRESCRIBED_CONC_HEAT_FLOW</code>	Node field	
<code>PRESCRIBED_CONC_SEEPAGE_FLOW</code>	Node field	
<code>PRESCRIBED_CONV_TEMPERATURE</code>	Node field	
<code>(ABC)-PRESCRIBED_DISPLACEMENT</code>	Node field	
<code>(XYZ)-PRESCRIBED_DISPLACEMENT</code>	Node field	
<code>(ABC)-PRESCRIBED_FLUID_TRACTION</code>	Node field	
<code>(XYZ)-PRESCRIBED_FLUID_TRACTION</code>	Node field	
<code>PRESCRIBED_FLUID_P0</code>	Node field	
<code>PRESCRIBED_FLUID_POTENTIAL</code>	Node field	
<code>(ABC)-PRESCRIBED_FORCE</code>	Node field	
<code>(XYZ)-PRESCRIBED_FORCE</code>	Node field	
<code>PRESCRIBED_MASS_RATIO_*</code>	Node field	
<code>(ABC)-PRESCRIBED_MOMENT</code>	Node field	
<code>(V1,V2)-PRESCRIBED_MOMENT</code>	Node field	
<code>(XYZ)-PRESCRIBED_MOMENT</code>	Node field	
<code>PRESCRIBED_OVALIZATION-(123456)</code>	Node field	
<code>PRESCRIBED_PRESSURE</code>	Node field	
<code>PRESCRIBED_RAD_TEMPERATURE</code>	Node field	
<code>(ABC)-PRESCRIBED_ROTATION</code>	Node field	
<code>(V1,V2)-PRESCRIBED_ROTATION</code>	Node field	
<code>(XYZ)-PRESCRIBED_ROTATION</code>	Node field	
<code>PRESCRIBED_TEMPERATURE</code>	Node field	
<code>PRESCRIBED_TOTAL_HEAD</code>	Node field	
<code>(ABC)-PRESCRIBED_VELOCITY</code>	Node field	
<code>(XYZ)-PRESCRIBED_VELOCITY</code>	Node field	
<code>PRESCRIBED_WARPING-(123456)</code>	Node field	

Table 13.1-18: Coordinate variables

Variable name	Variable location type	Notes
AREA_ELEDGE	Element face	11
AREA_ELFACE	Element face	12
AREA_ORIGINAL_ELEDGE	Element face	11
AREA_ORIGINAL_ELFACE	Element face	12
(XYZ)-COORDINATE	Coordinate	1
(XYZ)-COORDINATE_INCREMENT	Coordinate	2
(XYZ)-ORIGINAL	Coordinate	7
DIRECTOR_VECTOR-(XYZ)_CURRENT	Element local node field	9
DIRECTOR_VECTOR-(XYZ)_ORIGINAL	Element local node field	10
DISTANCE	Coordinate	3
DISTANCE_POSITION	Coordinate	3
(XYZ)-LEVER	Coordinate	4
(XYZ)-POSITION	Coordinate	5
SURFACE_NORMAL-(XYZ)	Coordinate	6
SURFACE_NORMAL-(XYZ)_ORIGINAL	Coordinate	8

1) These are the initial coordinates (coordinates of the original configuration of the model), when used with responses of type load-step. These are the reference coordinates (coordinates of the reference configuration), when used with responses of type mode-shape, response-spectrum, harmonic and random (recall that these responses include a reference time).

2) These give the changes in the X, Y and Z coordinates between the original configuration and the current configuration. They are used when the coordinates of the model change during the analysis, for example, in fracture mechanics crack propagation analysis.

3) These variables, when used in a line list or line graph, are defined as follows:

$$\text{DISTANCE} = \text{last computed DISTANCE} + (\text{distance between previous point and current point}),$$

$$\text{DISTANCE of first point} = 0.0$$

with a similar definition for DISTANCE-POSITION. (The difference between DISTANCE and DISTANCE-POSITION is that DISTANCE is computed based on

the original configuration of the model and `DISTANCE-POSITION` is computed based on the deformed configuration of the model.) These variables are path dependent and can only be used in a line list or line graph.

4) The lever variables are defined as follows: $x^L = (x - x_{REF})^N$ with similar definitions for the y and z lever variables. (x, y, z) are either the original or current coordinates, $(x_{REF}, y_{REF}, z_{REF})$ are the coordinates of a reference point and N is an integer greater than or equal to 1. The choice of original or current coordinate, $(x_{REF}, y_{REF}, z_{REF})$ and N are entered as part of a result control depiction.

The purpose of the lever variables is to make it easier to compute moments about a reference point.

5) These are the current coordinates (coordinates of the current configuration of the model), when used with responses of type load-step. These are the reference coordinates (coordinates of the reference configuration), when used with responses of type mode-shape, response-spectrum, harmonic and random (recall that these responses include a reference time).

6) These variables are used to access the face outwards normal of mesh integration or mesh extreme points.

These variables can also be used when plotting bands (for example, a resultant can contain these variables when plotting bands). When plotting bands, these variables refer to the normal at the current time (when the response is of type load-step), or at the reference time (when the response is of type mode-shape, response-spectrum, harmonic or random).

7) These are the initial coordinates (coordinates of the original configuration of the model), when used with any response type.

8) These variables can be used when plotting bands (for example, a resultant can contain these variables), except when plotting bands on cutting surfaces. When plotting bands, these variables refer to the normal of the original configuration of the model.

9) `DIRECTOR_VECTOR-(XYZ)_CURRENT` are the current components of the shell nodes director vectors. These variables are available from ADINA at the local nodes of shell elements.

10) `DIRECTOR_VECTOR-(XYZ)_ORIGINAL` are the original components of the shell nodes director vectors. These variables are available from ADINA at the local nodes of shell elements.

11) These are the current area and original area of an ADINA-F element edge (2D analysis). For planar analysis, unit thickness is assumed. For axisymmetric analysis, one radian is assumed.

12) These are the current area and original area of an ADINA-F element face (3D analysis).

Table 13.1-19: Thickness variables

Variable name	Variable location type	Notes
THICKNESS	Element local node field	1, 2
THICKNESS_CHANGE	Element local node field	1, 3
THICKNESS_CURRENT	Element local node field	1, 4
THICKNESS_ORIGINAL	Element local node field	1, 2
THICKNESS_STRAIN	Element local node field	1
THINNING	Element local node field	1

1) These variables are available from ADINA at the local nodes of ADINA shell elements.

2) THICKNESS and THICKNESS_ORIGINAL are the shell thicknesses in the original configuration of the model.

3) $THICKNESS_CHANGE = THICKNESS_CURRENT - THICKNESS_ORIGINAL$.

4) THICKNESS_CURRENT is the shell thickness in the current configuration. THICKNESS_CURRENT is different than THICKNESS and THICKNESS_ORIGINAL only for the top-bottom nodes of transition shell elements and for large strain shell elements.

5) THICKNESS_STRAIN is the natural logarithm of THICKNESS_CURRENT divided by THICKNESS_ORIGINAL.

6) $THINNING = -(THICKNESS_CHANGE / THICKNESS_ORIGINAL)$.

Table 13.1-20: Time variables

Variable name	Variable location type	Notes
LAMBDA	Loc-independent	1
LOAD_STEP	Loc-independent	

TIME	Loc-independent
TIME_FUNCTION_*	Loc-independent

1) LAMBDA is the load vector multiplier in the LDC method.

Table 13.1-21: Frequency/mode variables

Variable name	Variable location type	Notes
BUCKLING_LOAD_FACTOR	Loc-independent	1
EIGENVALUE	Loc-independent	2
FREQUENCY	Loc-independent	3
(XYZ)-MODAL_PARTICIPATION_FACTOR	Loc-independent	4
MODE_NUMBER	Loc-independent	
NATURAL_FREQUENCY	Loc-independent	5
EIGENSOLUTION_ERROR_MEASURE	Loc-independent	6

1) This is the buckling load factor computed in linearized buckling analysis.

2) This is the eigenvalue as computed by ADINA. The natural frequency is the square root of the eigenvalue.

3) The dimensions of frequency are cycles/(unit time)

4) These variables refer to modal participation factors for ground motions acting in the X, Y and Z directions.

5) The dimensions of natural frequency are radians/(unit time).

6) The eigensolution error measure is used to measure the accuracy of the computed eigenvalues, see Section 6.2.

Table 13.1-22: Contact variables

Variable name	Variable location type	Notes
CONTACT_STATE_FLAG	Node discrete	1
NODAL_CONTACT_GAP	Node field	2
NODAL_CONTACT_STATUS	Node field	3
NORMAL-(XYZ)	Contact	
SEGMENT_AREA	Contact	
SEGMENT_LENGTH	Contact	

- 1) The contact state flag has the following values: 0=unknown, 1=free, 2=sticking, 3=sliding. This variable is defined only for node to node contact.
- 2) NODAL_CONTACT_GAP is output when the “new” contact segments are used.
- 3) NODAL_CONTACT_STATUS is output when the “new” contact segments are used. It has the following values: =1, node is dead (on a dead contact segment), =2, node is open, =3, node is closed, frictionless contact, =4, node is closed, slipping contact, =5, node is closed, sticking contact.

Table 13.1-23: Energy variables

Variable name	Variable location type	Notes
KINETIC_ENERGY_DENSITY	Node field	1
STRAIN_ENERGY_DENSITY	Element field	

- 1) This is computed by the AUI as a predefined resultant using the formula

$$\text{KINETIC_ENERGY_DENSITY} = 0.5 * \text{ELEMENT_DENSITY} * (\langle \text{X-VELOCITY} \rangle^2 + \langle \text{Y-VELOCITY} \rangle^2 + \langle \text{Z-VELOCITY} \rangle^2)$$

You must define variable ELEMENT_DENSITY before using variable KINETIC_ENERGY_DENSITY.

Table 13.1-24: Miscellaneous variables

Variable name	Variable location type	Notes
CENTER_OF_MASS- (XYZ)	Loc-independent	1,2
CENTROID- (XYZ)	Loc-independent	1,3
COORDINATE_SYSTEM_NUMBER	Loc-independent	4
FLUID_DOF_TYPE	Node discrete	5
INERTIA-P1	Loc-independent	1,6
INERTIA-P1_DIRECTION- (XYZ)	Loc-independent	1,7

INERTIA-P2	Loc-independent	1,8
INERTIA-P2_DIRECTION- (XYZ)	Loc-independent	1,9
INERTIA-P3	Loc-independent	1,10
INERTIA-P3_DIRECTION- (XYZ)	Loc-independent	1,11
LINE_INDEX_NUMBER	Loc-independent	12
MASS	Loc-independent	1,13
MOMENT_OF_INERTIA- (XX, YY, ZZ)	Loc-independent	1,14
NUMBER_OF_SUBINCREMENTS	Element field	
PIPE_CROSS-SECTION_AREA	Node field	15
PIPE_SKIN_AREA	Node field	16
PRODUCT_OF_INERTIA- (XY, XZ, YZ)	Loc-independent	1,17
RADIOSITY	Node field	
REACTION_MOMENT- (XYZ)	Node discrete	22
ROTATIONAL_SYSTEM	Node discrete	18
SURFACE_MOMENT- (XYZ)	Mesh surface	21
TRANSLATIONAL_SYSTEM	Node discrete	19
VOLUME	Loc-independent	1,20

1) These variables are available for ADINA models when mass properties calculations are requested in ADINA-IN for all element groups in the model. They represent the given properties for the entire structure (or for that part of the structure selected in the AUI) and are therefore location-dependent.

2) This is the center of mass in the global coordinate system.

3) This is the centroid (center of volume) in the global coordinate system.

4) This variable stores the number of the coordinate system into which stresses/strains are transformed, see Section 13.1.3.

5) The fluid dof type has the following values: 0=free velocity potential, 1=fixed, 2=constrained, 3=free velocity potential, 4=free P_0 degree of freedom.

6) This is the largest principal inertia, given in a coordinate system with origin at the center of mass and axes parallel to the global coordinate system.

7) This is the principal axis associated with the largest principal inertia.

8) This is the intermediate principal inertia, given in a coordinate system with origin at the center of mass and axes parallel to the global coordinate system.

- 9) This is the principal axis associated with the intermediate principal inertia.
- 10) This is the smallest principal inertia, given in a coordinate system with origin at the center of mass and axes parallel to the global coordinate system.
- 11) This is the principal axis associated with the smallest principal inertia.
- 12) This is used when listing or graphing results along a line.
- 13) This is the total mass.
- 14) These are the moments of inertia, given in a coordinate system with origin at the center of mass and axes parallel to the global coordinate system. The XX component of the inertia tensor is equal to the XX moment of inertia.
- 15) This is the sum (over all pipe elements attached to the node) of the pipe internal cross-sectional area, based upon the current geometry.
- 16) This is the sum (over all pipe elements attached to the node) of the internal pipe skin area, based upon the current geometry.
- 17) These are the products of inertia, given in a coordinate system with origin at the center of mass and axes parallel to the global coordinate system. The XY component of the inertia tensor is the negative of the XY product of inertia.
- 18) Rotational system has the following values: 0=rotational results are stored in the global coordinate system, 1=rotational results are stored in a skew system, 2=rotational results are stored in a shell midsurface coordinate system (the shell node has five degrees of freedom).
- 19) Translational system has the following values: 0=translational results are stored in the global coordinate system, 1=translational results are stored in a skew system.
- 20) This is the total volume.
- 21) These are computed by the AUI as predefined resultants using the formulas

$$\begin{aligned}\text{SURFACE_MOMENT-X} &= \\ &\quad \langle Y\text{-LEVER} \rangle * \langle \text{SURFACE_TRACTION-Z} \rangle - \\ &\quad \langle Z\text{-LEVER} \rangle * \langle \text{SURFACE_TRACTION-Y} \rangle\end{aligned}$$

$$\begin{aligned}\text{SURFACE_MOMENT-Y} = & \\ & \langle \text{Z-LEVER} \rangle * \langle \text{SURFACE_TRACTION-X} \rangle - \\ & \langle \text{X-LEVER} \rangle * \langle \text{SURFACE_TRACTION-Z} \rangle\end{aligned}$$
$$\begin{aligned}\text{SURFACE_MOMENT-Z} = & \\ & \langle \text{X-LEVER} \rangle * \langle \text{SURFACE_TRACTION-Y} \rangle - \\ & \langle \text{Y-LEVER} \rangle * \langle \text{SURFACE_TRACTION-X} \rangle\end{aligned}$$

See Table 13.1-18 for the definitions of the lever variables and Table 13.1-10 for the definitions of the traction variables.

22) These are computed by the AUI as predefined resultants using the formulas

$$\begin{aligned}\text{REACTION_MOMENT-X} = & \\ & \langle \text{Y-LEVER} \rangle * \langle \text{Z-REACTION} \rangle - \langle \text{Z-LEVER} \rangle * \langle \text{Y-REACTION} \rangle\end{aligned}$$
$$\begin{aligned}\text{REACTION_MOMENT-Y} = & \\ & \langle \text{Z-LEVER} \rangle * \langle \text{X-REACTION} \rangle - \langle \text{X-LEVER} \rangle * \langle \text{Z-REACTION} \rangle\end{aligned}$$
$$\begin{aligned}\text{REACTION_MOMENT-Z} = & \\ & \langle \text{X-LEVER} \rangle * \langle \text{Y-REACTION} \rangle - \langle \text{Y-LEVER} \rangle * \langle \text{X-REACTION} \rangle\end{aligned}$$

See Table 13.1-18 for the definitions of the lever variables and Table 13.1-11 for the definitions of the reaction variables.

Table 13.1-25: User-defined variables

Variable name	Variable location type	Notes
INT_USER_VARIABLE_*	Element field	1
ONE	Loc-independent	2
USER_VARIABLE_*	Element field	1

- 1) These variables are used in conjunction with the user-supplied material model.
- 2) This variable has the constant value “1”. It is included so that you can determine the total area of a domain that is defined as a mesh integration point.
- 3) Resultants that you define in the AUI are added to the list of user-defined variables. The variable location type of a resultant is determined by the variables that define the resultant.

-
- The AUI can transform node field and node discrete results
-

calculated by ADINA in one coordinate system to another coordinate system as shown in Table 13.1-26.

Table 13.1-26: Coordinate transformations for node field and node discrete variables performed by the AUI

Coordinate system of ADINA/ADINA-F result	Variable prefix		
	X,Y,Z	A,B,C	V1,V2
Global	1	1	3
Skew	2	1	3
V1,V2	3	1	1

- 1) The result is computed without any coordinate transformation.
- 2) The result is computed using a transformation from the skew system to the global system.
- 3) No result is computed.

Displacement and stress transformations: The AUI can transform displacements, stresses, strains and similar quantities to user-defined coordinate systems, including cylindrical and spherical systems.

In order to activate this feature, you set the coordinate system of the result control dialog box to the number of the coordinate system that you define. By default, the coordinate system is 0 (the global Cartesian system).

You access the transformed displacements using variables 1-DISPLACEMENT, 2-DISPLACEMENT, 3-DISPLACEMENT, and the transformed stresses using variables STRESS-11, STRESS-22, STRESS-33, STRESS-12, STRESS-13, STRESS-23. Other transformed quantities are accessed using similar names. These variables can be used in any plotting or listing command.

The AUI evaluates the transformed stress and strain variables from the stresses and strains in the global system, as computed by the solution program. Therefore the stresses and strains must have been calculated in the global system by the solution program.

The direction numbers in the variable names 1 -

DISPLACEMENT, STRESS-11, STRESS-22, etc. correspond to coordinate directions as shown in the following table:

System type	Direction 1	Direction 2	Direction 3
Cartesian	XL	YL	ZL
Cylindrical	R	THETA	XL
Spherical	R	THETA	PHI

Thus, by default, STRESS-11 = STRESS-XX, STRESS-22 = STRESS-YY, etc.

You can plot the directions used for displacement and stress transformations in a mesh plot. This is done from within the Element Depiction dialog box reached using the Modify Meshplot icon. This feature is useful for verifying that the displacement and stress transformation directions are correct.

Please note:

You can smooth the transformed stresses. The smoothing is done in the global coordinate system and the smoothed stresses are transformed into the user-specified system.

The transformed off-diagonal strain components (STRAIN-12, STRAIN-13, STRAIN-23) are engineering quantities, not tensorial quantities.

You can use the variables STRESS-11, STRESS-22, etc. within resultants.

13.1.2 Where results can be obtained

- Results can be evaluated at various types of locations within the model.

Location-independent: Used for variables that are location-independent, see above.

Node: locations at nodes.

Element/layer: locations within elements or element layers. These can be the integration points, the element local nodes or other locations within the element.

To specify the element, you need to enter the element group

number and element number. If the element is an ADINA multilayer shell element, you also need to enter the layer number. Otherwise you can enter a layer number of 1.

To specify the point within the element (or element layer), there are several conventions, as follows. In ADINA system elements, there are two subtypes of element points: integration points and local nodes. In any given analysis, for any given element, the results are computed by the solution program at only one of these subtypes.

In the AUI, you can request that the results be evaluated at the element points where they were computed by the solution program, or you can request that the results be extrapolated to other element points. This latter option is available for some elements as shown in Table 13.1-25.

Table 13.1-25 Extrapolation of element results
ADINA

Element type	Result location	Dimension for results	Can extrapolate to
Truss	Int. pts.	1-D	$-1.0 \leq r \leq 1.0$
2-D solid or fluid	Int. pts.	2-D	$-1.0 \leq r, s \leq 1.0^*$
3-D solid or fluid	Int. pts.	3-D	$-1.0 \leq r, s, t \leq 1.0^*$
Iso-beam (3-D)	Int. pts.	3-D	$-1.0 \leq r, s, t \leq 1.0$
Iso-beam (2-D)	Int. pts.	2-D	$-1.0 \leq r, s \leq 1.0$
Plate	Int. pts.	2-D	$0.0 \leq r, s \leq 1.0$
Shell	Int. pts.	3-D	$-1.0 \leq r, s, t \leq 1.0$
ADINA-T			
Element type	Result location	Dimension for results	Can extrapolate to
1-D	Int. pts.	1-D	$-1.0 \leq r \leq 1.0$
2-D conduction	Int. pts.	2-D	$-1.0 \leq r, s \leq 1.0^*$
3-D conduction	Int. pts.	3-D	$-1.0 \leq r, s, t \leq 1.0^*$
Shell conduction	Int. pts.	3-D	$-1.0 \leq r, s, t \leq 1.0$
ADINA-F:			
Element type	Result location	Dimension for results	Can extrapolate to
2-D fluid	Int. pts.	2-D	$-1.0 \leq r, s \leq 1.0^*$
3-D fluid	Int. pts.	3-D	$-1.0 \leq r, s, t \leq 1.0^*$

*) If the element is triangular or tetrahedral, the element uses triangular/tetrahedral area coordinates as the isoparametric coordinates and the coordinates range from 0.0 to 1.0.
(end of table)

You use the Defined By parameter and its related parameters in the element result point dialog box to select the element point as follows:

Grid point: This option is available only if results were calculated at the element integration points by the solution program.

The element conceptually contains an evenly spaced grid in the elements' isoparametric system. The grid can have one, two or three dimensions depending on the element type.

To specify a grid point using a 1-D grid, give a two digit number AB: digit A gives the total number of grid divisions and digit B gives the desired grid point.

The isoparametric coordinate corresponding to the grid point is computed using

$$C = C_{\min} + \frac{B-1}{A-1} (C_{\max} - C_{\min})$$

where C_{\min} and C_{\max} are the minimum and maximum isoparametric coordinates that the elements can contain. For example, AB=52 corresponds to $r = -1/2$ for elements in which r can be between -1 and 1.

To specify a grid point using a 2-D grid, give a four digit number ABCD: digits A and B give the desired grid point in the r direction and digits C and D give the desired grid point in the s direction. For example, ABCD=2121 corresponds to $r = -1$, $s = -1$ and ABCD=3243 corresponds to $r = 0$, $s = 1/3$.

To specify a grid point using a 3-D grid, give a six digit number ABCDEF: digits A and B give the desired grid point in the r direction, digits C and D give the desired grid point in the s direction and digits E and F give the desired grid point in the t direction.

You can also specify the element centroidal point by giving a grid value of 0.

Label point: You specify the label point in the element at which results were computed by the solution program.

If the results were calculated at the integration points, then the label is a 1, 2, 3 or 4 digit number giving the integration point as shown in Table 13.1-26.

If the results were calculated at the element local nodes, then the label is the local node number.

No extrapolation is performed by the AUI when using the label option.

Table 13.1-26: Label point numbering

ADINA:

Element type	Result locations	Label number
Truss	Int. pts.	(INR)
2-D solid or fluid	Int. pts.	$10(\text{INR}) + \text{INS}^*$
3-D solid or fluid	Int. pts.	$100(\text{INR}) + 10(\text{INS}) + \text{INT}^*$
Hermitian beam	Int. pts.	$100(\text{INR}) + 10(\text{INS}) + \text{INT}$
Iso-beam (3-D)	Int. pts.	$100(\text{INR}) + 10(\text{INS}) + \text{INT}$
Iso-beam (2-D)	Int. pts.	$10(\text{INR}) + \text{INS}$
Plate	Int. pts.	(ITRI)
Shell (rectangular)	Int. pts.	$100(\text{INR}) + 10(\text{INS}) + \text{INT}$
Shell (triangular)	Int. pts.	$10(\text{ITRI}) + (\text{INT})$
Pipe	Int. pts.	$1000(\text{INA}) + 10(\text{INB}) + \text{INC}$
Spring	----	1
General	Stress transformation matrix	(J)

ADINA-T:

Element type	Result locations	Label number
1-D	Int. pts.	(INR)
2-D conduction	Int. pts.	$10(\text{INR}) + \text{INS}^*$
3-D conduction	Int. pts.	$100(\text{INR}) + 10(\text{INS}) + \text{INT}^*$
Shell (conduction)	Int. pts.	$100(\text{INR}) + 10(\text{INS}) + \text{INT}$

ADINA-F:

Element type	Result locations	Label number
2-D fluid	Int. pts.	$10(\text{INR}) + \text{INS}^*$
3-D fluid	Int. pts.	$100(\text{INR}) + 10(\text{INS}) + \text{INT}^*$

Notes:

Each quantity in parentheses represents one integer:

(ITRI) – integration point number in triangular integration

(ITET) - integration point number in tetrahedral integration

(INR), (INS), (INT) – integration point numbers for r, s and t coordinates

(INA), (INB), (INC) – integration point numbers for a, b and c coordinates

(J) – stress component number

*) If the element is triangular, the label number is (ITRI); if the element is tetrahedral, the label number is (ITET).

(end of table)

Node: This option is available only if results were calculated at the element integration points by the solution program.

You enter the global node number of a node connected to the element.

The results are extrapolated to the point within the element corresponding to the position of the node in the element. For example, if the first node of a 2-D element is number 65, then selecting node 65 is equivalent to selecting the point $(r,s) = (1,1)$ in the element.

It is necessary to specify the element number when this option is used because, when the results are unsmoothed, the results at a node point are different from the different elements attached to the node.

The use of this option does not, by itself, smooth results from neighbor-ring elements to the specified node. But if smoothing is turned on, then any element that contains the specified node will give the same results (because the smoothed results are continuous between the elements).

Isoparametric coordinates: This option is available only if results were calculated at the element integration points by the solution program.

You directly enter the r, s and t coordinates of the desired point. If the element is 1-D or 2-D, the unused coordinates are ignored.

Special considerations for the ADINA shell element: The element is considered to be a 3-D element for the purpose of specifying element points. You can specify that the element point lies on the shell midsurface using the following options:

Node number: Specify the node number of a midsurface node.

Grid location: Specify a grid location of 0, or a grid location in which the 5th and 6th digits are equal to 1.

When the element point lies on the shell midsurface, when the results extrapolated from the element integration points are evaluated at the point, the t coordinate and layer used for the evaluation are selected by the parameters of the result control depiction.

Element section: locations within element sections (ADINA only, beam and shell elements). These can be the section integration points, the element local nodes or other locations within the element section.

Two types of elements can have results output at element sections. The Hermitian beam element can have results output on the neutral line of the element; the neutral line is considered to be the element section. The ADINA shell element can have results output on the neutral surface of the element; the neutral surface is considered to be the element section.

Element section results are very similar to element result points. You can request that results be evaluated at the locations at which they were computed by ADINA or that the results be extrapolated to other locations using the grid, node and isoparametric coordinate parameters. It is important to remember the correct dimension of the element section: the section is a line for beam elements and a surface for shell elements. Hence, for example, a grid location is of the form AB for beam element section points and of the form ABCD for shell element section points.

Note that it is possible to list shell element stresses at shell element section result points if shell midsurface calculations are activated in the result control depiction.

Contact segments: (ADINA only). Locations in contact segments.

A contact surface contains one or contact segments, each of which contains a single location where ADINA computes results.

When you request the coordinates of a contact segment, the coordinates of the center of the segment are used.

Within a contact surface, contact segments are numbered sequentially starting from 1. Hence, to specify a contact segment result point location, you must specify three numbers: the contact group number, the contact surface number and the contact segment number.

Note that no results are available for contact segments on target contact surfaces. Those contact results that are output at nodal points (for example, concentrated contact forces) are considered by the AUI to be associated with node points.

Draw bead segments: (ADINA only). Locations in draw bead segments. A contact surface optionally contains one or more drawbead segments, each of which contains a single location where ADINA computes results.

Within a contact surface, draw bead segments are numbered sequentially starting from 1. Hence, to specify a draw bead segment result point location, you must specify three numbers: the contact group number, the contact surface number and the draw bead segment number.

Radiosity segments: Locations in radiosity segments. A radiosity surface contains one or more radiosity segments, each of which contains integration points at which the results are computed..

The options for entering locations within radiosity segments are very similar to those used for entering locations within element layers. You can request that results be evaluated at the locations where they were computed or that the results be extrapolated to other locations within the segment using the grid, node and isoparametric coordinate parameters. It is important to remember the correct dimensions of the radiosity segment. Hence, for example, a grid location is of the form AB for segments within radiosity lines and of the form ABCD for segments within radiosity surfaces.

Virtual shifts: (ADINA only). In order for you to access the results associated with virtual shifts in ADINA fracture mechanics analysis, the AUI allows you to define a result point at a virtual

shift. (In this case, the virtual shift does not occupy a point in space.)

Line contours: (ADINA only). In order for you to access the results associated with J-integral line contours in ADINA fracture mechanics analysis, the AUI allows you to define a result point at a J-integral line contour. (In this case, the line contour does not occupy a point in space.)

Element faces: (ADINA-F only). Results output at individual element edges (2D analysis) and element faces (3D analysis) by ADINA-F are generally most useful when they are summed on model boundaries, or other selected areas of the model. The AUI uses the element edge-set and element face-set concepts in order to group the element edges and faces. For example, in order to calculate the mass flux over a selected area of the model,

- 1) Create an element edge-set (2D analysis) or element face-set (3D analysis) corresponding to the selected area (from the command-line, use the commands ELEDGESET or ELFACESET). In many cases ADINA-F automatically creates useful element edge-sets and element face-sets.
- 2) Create a model result point of type element edge-set or element face-set (from the command-line, use the commands ELETPOINT or ELFSETPOINT).
- 3) List or plot the mass flux at the model result point. The output mass flux is the sum of the mass fluxes at all of the element edges or faces of the element face-set or edge-set.

You can also list the individual mass fluxes at the element edges or faces of an element edge-set or element face-set. For example, given an element edge-set or face-set:

- 1) Create a model result line of type element edge-set or element face-set (from the command-line, use the commands ELETLINE or ELFSETLINE).
- 2) List the mass flux on the model result line.

Mesh integration points: If you want to integrate a quantity over the faces of a mesh plot, define a mesh integration point.

The mesh integration point consists of: 1) a domain over which the integration is to take place and 2) the rules for performing the integration.

Domain: The domain is described by the name of a mesh plot and the name of a zone as follows. If the mesh plot does not have a cutting surface, the domain is the external faces of the meshplot (the “skin” of the meshplot). If the mesh plot has a cutting surface, the domain is the cutting surface faces.

Faces can be, for example, the boundaries of 3-D elements, the midsurfaces of shell or plate elements, or the interiors of 2-D elements. They cannot be the boundaries of shell or 2-D elements.

Another way to think of the domain is in terms of a band plot on the mesh plot. The domain is that area of the mesh plot that is covered by the bands.

Use the zone name to restrict the domain. A face is in the domain only if its associated element is in the specified zone.

Rules: The rules are the integration type, a multiplying factor, an integration order and the configuration over which the integration is performed. Let z be the integrand and S be the domain. Then the integration types are

$$\text{Integral:} \quad \int_S z dS$$

$$\text{Averaged:} \quad z_{av} = \frac{\int_S z dS}{S}$$

$$\text{Mean-square:} \quad z_{ms} = \frac{\int_S z^2 dS}{S}$$

Root-mean-square: $\sqrt{\frac{\int_S z^2 dS}{S}}$

Variance: $\frac{\int_S (z - z_{av})^2 dS}{S}$

Standard deviation: $\sqrt{\frac{\int_S (z - z_{av})^2 dS}{S}}$

Relative variance: $\frac{\int_S \left(\frac{z - z_{av}}{z_{av}} \right)^2 dS}{S}$

Relative standard deviation: $\sqrt{\frac{\int_S \left(\frac{z - z_{av}}{z_{av}} \right)^2 dS}{S}}$ (also referred to as the coefficient of variation)

Use the multiplying factor to multiply the result after the indicated integration is performed. Use the integration order to indicate the numerical integration order over each element face. Use the configuration to indicate whether to perform the integration in the original configuration of the model or in the current deformed configuration.

Please note that you do not specify the integrand within the mesh integration point. You specify the integrand when you reference the mesh integration point within another dialog box.

Once you have defined a mesh integration point, you can use it within any command that accepts a model result point. For example, you can plot the time history of an integrated quantity using Graph→Response Curve (Model Point).

There are certain variables and predefined resultants that can only be used in conjunction with mesh integration (or mesh

extreme) points. These are all based upon the face outwards normal. The face outwards normal can be accessed through variables SURFACE_NORMAL-X, SURFACE_NORMAL-Y, SURFACE_NORMAL-Z. Useful quantities for which predefined resultants are defined are VOLUME_FLUX_SURFACE, MASS_FLUX_SURFACE, SURFACE_TRACTION-X, SURFACE_TRACTION-Y, SURFACE_TRACTION-Z, MOMENTUM_FLUX-X_SURFACE, MOMENTUM_FLUX-Y_SURFACE, MOMENTUM_FLUX-Z_SURFACE, KINETIC_ENERGY_DENSITY, KINETIC_ENERGY_FLUX_SURFACE, HEAT_FLUX_SURFACE. Any of these variables can be used along with a mesh integration point. For example, you can determine the volume flux of flow through a pipe by defining a mesh integration point corresponding to a cutting surface, then graphing or listing the variable VOLUME_FLUX at the mesh integration point.

You can also obtain the surface area of the mesh integration point by evaluating variable ONE at the point (variable ONE is defined to have the value 1.0).

You can also obtain the approximate volume of the mesh by use of the divergence theorem (also known as Gauss' theorem or Green's theorem), which is:

$$\int_V \nabla \cdot \mathbf{A} \, dV = \int_S \mathbf{A} \cdot \mathbf{n} \, dS$$

Simply define resultant ADOTDS to be <X-COORDINATE>* <SURFACE_NORMAL-X> and evaluate it at a mesh integration point. This works because $\nabla \cdot (<X-COORDINATE>, 0, 0) = 1$.

Mesh extreme points: If you want to determine the extreme value of a quantity over the faces of a mesh plot, define a mesh extreme point.

The mesh extreme point consists of: 1) a domain over which the search for the extreme value is to take place and 2) the rules for determining the extreme value.

Domain: The domain is defined in exactly the same way as for a mesh integration point, see above.

Rules: The rules are the extreme type, the multiplying factor, the search grid order and the domain configuration. The extreme types can be absolute maximum, maximum or minimum. Use the multiplying factor to multiply the result after the extreme value is found. Use the search grid order to indicate the number of sampling points within each element face. Use the domain configuration to indicate whether to search for the extreme value using the original or current deformed configuration of the model.

Please note that you do not specify the quantity within the mesh extreme point. You specify the quantity when you reference the mesh extreme point within another dialog box.

Once you have defined a mesh extreme point, you can use it within any command that accepts a model result point. For example, you can plot the time history of an extreme quantity using Graph→Response Curve (Model Point).

The variables that can be used for mesh integration points can also be used for mesh extreme points.

- Not all results can be obtained at all result locations. Table 13.1-27 shows the result location types for each type of available variable. Notice that certain combinations are allowed only when smoothing is requested (see below).

13.1.3 How results are evaluated (including smoothing)

- If a node field variable is requested within an element, the nodal shape functions are employed to interpolate the variable within the element.
- If an element/layer variable is requested within an element at a point, the variable is interpolated or extrapolated as follows:

RST interpolation: If the variable represents a floating-point value, the variable is interpolated or extrapolated using the nearest integration points and bilinear interpolation. If the variable represents an integer value, the value of the variable at the nearest integration point is used.

Table 13.1-27: Locations at which available variables can be evaluated

Variable location type	Result location types
Location-independent	All
Node field	Node, element/layer, contact segment, radiation segment
Element/layer field	Node (smoothing only), element/layer, section
Contact segment	Contact segment
Drawbead segment	Drawbead segment
Section field	Node (smoothing only), section
Radiation field	Node (smoothing only), radiosity segment
Line contour	Line contour
Virtual shift	Virtual shift
Node discrete	Node
Element local node	Element/layer (at local nodes only)
Element local node field	Element/layer, section
Coordinate	Node, element/layer, section, contact segment, radiation segment
Element face	Element edge-set or element face-set

Face interpolation: For 2D elements, this is the same as centroid interpolation, see below. For 3D elements, the closest element face is determined for the point. Then the results are extrapolated using RST interpolation to the center of the closest element face. Therefore the extrapolated results for all points in the element that are closest to a given element face are the same as the extrapolated element face center value.

Centroid interpolation: The results are extrapolated using RST interpolation to the element centroid. Therefore the extrapolated results at all points in the element are the same as the extrapolated centroid value.

Integration point interpolation: The results at the nearest integration point are used.

The choice of interpolation is made within the result control dialog box. RST interpolation is the default.

Smoothing

- The element/layer field or section results can be smoothed (that is, made continuous between adjacent elements or element sections). The smoothing process is now described in detail for element results; the smoothing process is similar for element section results.

Smoothing consists of two steps:

1) Smoothing to the node points. For each node,

a) Within each element attached to the node, the element integration point results are extrapolated to the node point. The extrapolation process is exactly the same as used when element integration point results are requested at other points within the element.

b) The contributions of each element are combined into a single result, as specified by the type of smoothing:

Averaged: averages

Minimum: minimum result is taken

Maximum: maximum result is taken

Difference: difference between the maximum and minimum result is taken

Extreme: extreme result (result furthest from 0.0) is taken

Error: Same as difference, except that only the element corner nodes are considered and the result is divided by a reference value that you enter (the default reference value is 1.0).

2) Extrapolation to the requested point. The extrapolation is exactly the same as used when nodal point results are requested within the elements. If the requested point is a nodal point, no action is taken in this step.

- Normally, all elements attached to a node contribute results to the node. However, you can choose to smooth results using only selected elements. This is done by specifying a zone name in the smoothing depiction. In this case, only elements within the zone contribute results to the node.

This option can be important when, for example, the model has several materials and you do not want to smooth stresses across the material interfaces.

- The following restrictions for smoothing are important to remember:
 - ▶ Results from 2-D, 3-D (solid or fluid) and plate elements can be smoothed. Results from shell elements can be smoothed on the midsurface of the shell elements if shell element midsurface calculations are turned on in the result control depiction.
 - ▶ You must have requested that results be saved at the element integration points during preprocessing.
 - ▶ Results that evaluate as integer values, such as the crack flag from the concrete model, can be smoothed into non-integer values, even when the non-integer values have no meaning.
 - ▶ Smoothing results across element groups may not be physically appropriate, for example, when the elements groups have different material properties.
 - ▶ Results are smoothed component-by-component without taking into account possible differences in coordinate systems between adjacent elements. This is especially important for results from shell elements.
- Control of smoothing is done by choosing a smoothing depiction, or by defining and using a smoothing depiction. By

default, smoothing is turned off in the AUI.

Plotting results onto midsurfaces of shells: The AUI allows you to specify which shell layer and t coordinate value to use when plotting results onto shell midsurfaces. This allows you to plot, for example, the results from the top of the shells onto the shell midsurfaces. You can also choose to plot results evaluated at the shell midsurface onto the shell midsurface. These choices are made using a result control depiction. By default, results from the top of the shell are presented on the shell midsurface.

13.1.4 When (for which load step, etc.) are results available

- Results available from solution programs of the ADINA system are

Load step results: Results associated with load steps (time steps). Each load step has a unique load step number and solution time.

Mode shape results: Results associated with mode shapes. Each mode shape has two numbers that define it uniquely, the mode shape number and the solution time corresponding to the solution from which the mode shapes were calculated.

Residual results: Results associated with residual (static correction) responses in response spectrum analysis (see Section 9.1 for theoretical background).

- Corresponding to these three types of results, the AUI has three response types:

Load-step: Allows you to reference results from a specific load step. A load step is referenced by its solution time.

If you enter a solution time between two solution times for which results are output by the solution program, the AUI automatically interpolates the results to the entered solution time. This is an example of a time transformation discussed in Section 13.1.

Mode-shape: Allows you to reference results from a specific

mode shape. A mode shape is referenced by its mode shape number and solution time.

Residual: Allows you to reference residual results. Residual results are referenced by the ground motion direction and solution time corresponding to the solution for which the residual results were obtained.

- The AUI includes additional response types used in various analyses.

Response spectrum: A calculation method used for computing results in response spectrum analysis. You specify the modal combination method, loading spectrum, modal damping table and other information. See Section 9.1 for theoretical background.

Harmonic: A calculation method used for computing results in harmonic vibration analysis. You specify the desired harmonic response (maximum, phase angle, rms amplitude or signed amplitude), the loading frequency, the loading spectra, the damping table and other information. See Section 9.3 for theoretical background.

Random: A calculation method used for computing results in random vibration analysis. You specify the desired random response (rms amplitude or power-spectral-density), the loading frequency (for power-spectral-density only), the loading spectra, the damping table and other information. See Section 9.4 for theoretical background.

Response-combination: A calculation method for combining previously defined responses. These responses can be of any type except for response-combination and any mixture of the allowed response types can be used. For each previously defined response, you can specify a weighting factor and a combination method.

The AUI uses the following algorithm to evaluate the response-combination:

accum=0.0

```
For (each response in the response-combination) {  
  value=(result for the response) * multiplying factor  
  if (method = algebraic) {  
    accum=accum + value  
  }  
  else if (method = abs) {  
    accum=abs(accum) + abs(value)  
  }  
  else if (method = srss) {  
    accum=sqrt(accum**2 + value**2)  
  }  
  else if (method = signed 1) {  
    if (value >= 0.0) {  
      accum=value + abs(accum)  
    }  
    else {  
      accum=value - abs(accum)  
    }  
  }  
  else if (method = signed 2) {  
    if (value >= 0.0) {  
      accum=accum + abs(value)  
    }  
    else {  
      accum=accum - abs(value)  
    }  
  }  
}
```

You might use methods signed 1 and signed 2, for example, when including a static load case along with the combined responses for different spatial directions in response spectrum analysis.

Envelope: At a point in the model, a given result (such as the value of a stress component) has a time history, and the AUI can perform a calculation on the time history, for example, the AUI can obtain the maximum value and the corresponding solution time. Of course, the magnitude of the maximum value and the corresponding solution time is different from point to point. The envelope of the result is a listing of all of the calculated results; the listing gives the model points, the calculated results

and the corresponding solution times.

To obtain an envelope listing in the AUI, define a response of type envelope. This response associates a response name with a calculation method in which several responses are evaluated, a calculation is made on the responses and one value is returned. The type of calculated value can be an extreme value, a time-integral value or a sampling-based value:

Extreme value: The minimum, maximum, absolute value of the maximum or difference between maximum and minimum.

$$\text{Time integral: } \int_{t_1}^{t_2} v(t) dt$$

$$\text{Time averaged: } v_{tav} = \frac{\int_{t_1}^{t_2} v(t) dt}{t_2 - t_1}$$

$$\text{Time mean-square: } v_{tms} = \frac{\int_{t_1}^{t_2} v(t)^2 dt}{t_2 - t_1}$$

$$\text{Time root-mean-square: } \sqrt{v_{tms}}$$

$$\text{Time variance: } v_{tvar} = \frac{\int_{t_1}^{t_2} (v(t) - v_{tav})^2 dt}{t_2 - t_1} = v_{tms} - v_{tav}^2$$

$$\text{Time standard deviation: } v_{tsd} = \sqrt{v_{tvar}}$$

$$\text{Time relative variance: } v_{trvar} = \frac{\int_{t_1}^{t_2} \left(\frac{v(t) - v_{tav}}{v_{tav}} \right)^2 dt}{t_2 - t_1} = \frac{v_{tvar}}{v_{tav}^2}$$

Time relative standard deviation: $v_{trsd} = \sqrt{v_{trvar}}$

Sample sum: $\sum_{i=1}^N v_i$

Sample averaged: $v_{sav} = \frac{\sum_{i=1}^N v_i}{N}$

Sample mean-square: $v_{sms} = \frac{\sum_{i=1}^N v_i^2}{N}$

Sample root-mean-square: $\sqrt{v_{sms}}$

Sample variance: $v_{svar} = \frac{\sum_{i=1}^N (v_i - v_{sav})^2}{N} = v_{sms} - v_{sav}^2$

Sample standard deviation: $v_{ssd} = \sqrt{v_{svar}}$

Sample relative variance: $v_{srvar} = \frac{\sum_{i=1}^N \left(\frac{v_i - v_{sav}}{v_{sav}} \right)^2}{N} = \frac{v_{svar}}{v_{sav}^2}$

Sample relative standard deviation: $v_{srstd} = \sqrt{v_{srvar}}$

In these formulas, $v(t)$ is the time history of a solution quantity and v_i is the value of a solution quantity.

There are two ways to select the responses included in the envelope:

Range: The AUI evaluates the value for a range of load-steps and returns the extreme value. You can choose the start

and end times, the intermediate steps to sample, and whether to interpolate results to steps for which results are not saved.

Selected: The AUI evaluates the value at each response specified in the data input lines of this command and returns the extreme value. Time-integral values cannot be evaluated in this case.

The AUI performs the calculation independently at each requested point in the model.

- The AUI includes dialog boxes for each of the response types. Each dialog box defines a response of the desired response type. You then use the response in another command when you want to plot or list the results using that response.

For response types response spectrum, harmonic, random, response combination, random or envelope, the AUI does not actually compute the response until the response is referenced by another command.


13.2 Evaluation of meshes used

- Once the solution with a given finite element mesh has been obtained, you may want to identify whether the mesh used has been fine enough. This can be achieved in the AUI by plotting bands of quantities that are the derivatives of the primary element variables.
- For example, a band plot of pressures (or effective stress or any stress variable) can be produced). The unsmoothed or smoothed pressures can be displayed. Consider that unsmoothed pressures are plotted. If the stresses are to be continuous in the exact solution (which may not be the case when there are discontinuities in material properties), the pressure bands will be reasonably continuous if the mesh has been fine enough. The breaks in bands indicate the severity of the stress discontinuities.

ref. T. Sussman and K.J. Bathe, "Studies of finite element procedures – stress band plots and the evaluation of finite element meshes," *Engineering Computations*, Vol. 3, No. 3, pp. 178-191, September 1986.

- The breaks in the bands are easily seen if the bands are drawn using two colors that alternate (so that the resulting plot looks like a photoelastic image). This option is available in the AUI when a band table of type repeating is requested.
- Another error indicator based on jumps in results between elements when too coarse a mesh has been used is the maximum difference between unsmoothed results at nodes. This error indicator is obtained by using a smoothing depiction of type error; see Section 13.1.3 for a description of the smoothing depictions.

This error measure can be divided by a reference value.

For convenience, the AUI has an Error Plots icon  that allows you to easily plot an error indicator. For example, suppose that a band plot of unsmoothed pressure is displayed. When the Error Plots icon is clicked,

- 1) The maximum pressure in the unsmoothed band plot is taken as

the error reference value.

2) At each node, the maximum difference between unsmoothed pressures is computed. This maximum difference divided by the error reference value is the nodal error indicator.

3) The nodal error indicator is then drawn as a band plot.

- See the AUI Primer, Problem 13, for an example showing the evaluation of meshes using band plot.

Index

2

- 2-D solid elements, [17](#)
- 3-D plane stress, [17](#)
- axisymmetric, [17](#)
- element output, [29](#)
- formulations, [24](#)
- generalized plane strain, [17](#)
- mass matrices, [28](#)
- material models, [24](#)
- numerical integration, [26](#)
- plane strain, [17](#)
- plane stress, [17](#)
- recommendations for use, [36](#)
- triangular, [23](#)
- 2nd Piola-Kirchhoff stresses, [459](#)

3

- 3-D solid elements, [40](#)
- element output, [53](#)
- formulation, [49](#)
- mass matrices, [53](#)
- material models, [48](#)
- numerical integration, [50](#)
- prisms, [47](#)
- pyramids, [47](#)
- recommendations for use, [62](#)
- tetrahedra, [47](#)
- 3D-iterative solver, [1150](#)
- 3D-shell elements, [232](#)

5

- 5 degrees of freedom node, [191](#)

6

- 6 degrees of freedom node, [193](#)

A

- Absolute contact damping, [873](#), [997](#)
- Acceleration variables, [1513](#)
- Accumulated effective plastic strain, [494](#), [530](#)
- ADINA heat transfer capabilities, [1487](#)
- ADINA System documentation, [5](#)
- ADINA-TMC
 - Piezoelectric analysis, [1496](#)
 - soil consolidation analysis, [1493](#)
- Alignment elements, [343](#)
 - element output, [385](#)
- Analysis zooming, [1449](#)
- Analytical rigid targets, [912](#)
- Anand material model, [756](#)
- Arc length method, [1167](#)
- Arrival time, [1048](#)
- Arruda-Boyce material model, [641](#)
 - 3-D analysis, [642](#)
 - axisymmetric analysis, [642](#)
 - plane strain analysis, [642](#)
 - plane stress analysis, [642](#)
 - selection of material constants, [642](#)
- ATS method, [1047](#), [1160](#), [1176](#)
 - dynamic analysis, [1163](#)
 - low speed dynamics, [1158](#)
- Automatic step incrementation, [1153](#)
 - ATS method, [1160](#)
 - LDC method, [1167](#)
- Axisymmetric shell elements, [165](#), [167](#), [175](#)
 - element output, [173](#), [176](#)
 - formulations, [174](#)
 - linear, [171](#)
 - mass matrices, [176](#)
 - material models, [174](#)
 - nonlinear, [174](#)

numerical integration, [169](#), [175](#)

B

Bathe method, [1487](#)

Bathe subspace iteration method, [1137](#)

Beam element

cross-section, [68](#)

Beam elements, [63](#)

2-D action, [105](#)

3-D action, [105](#)

auxiliary node, [66](#)

box cross-section, [122](#)

coefficient of thermal expansion,
[103](#)

cross-section geometric properties,
[70](#)

degrees of freedom, [66](#)

displacements, [72](#)

end-release option, [155](#)

fixed-end force correction, [108](#)

forces/moments, [66](#)

general cross-section, [114](#)

geometry, [66](#)

I cross-section, [127](#)

kinetic energy, [84](#)

L cross-section, [137](#)

large displacement formulation, [97](#)

mass matrices, [101](#)

modeling hints, [160](#)

moment curvature, [144](#)

nodal point force output, [106](#)

nonlinear elasto-plastic, [103](#)

numerical integration, [104](#)

orientation vector, [66](#)

pipe cross-section, [119](#)

rectangular cross-section, [114](#)

rigid-end option, [159](#)

section force output, [106](#)

shear center, [69](#)

standard beam, [93](#)

stiffness matrix, [102](#)

strains, [74](#)

stress output, [106](#)

stresses, [75](#)

torsional response, [75](#)

U cross-section, [132](#)

Wagner effect, [79](#)

warping beam, [94](#)

warping degrees of freedom, [162](#)

warping function, [85](#)

Bergstrom-Boyce viscoelasticity, [682](#)

BFGS method, [1153](#), [1155](#)

Bimoment, [76](#)

Bishear, [76](#)

Bolt option, [1455](#)

Boundary conditions

displacement, [1045](#)

essential, [1045](#)

force, [1045](#)

moment, [1045](#)

natural, [1045](#)

C

Cable elements, [9](#)

Cam-clay material model, [716](#)

initial stiffness, [719](#)

initial stresses, [718](#)

Cast iron material model, [784](#)

Cauchy stresses, [455](#), [459](#)

Centers of mass

inertial properties, [1437](#)

Centrifugal loads, [1071](#)

Centroids

inertial properties, [1437](#)

Cohesive elements, [388](#)

element output, [392](#)

Component mode synthesis (CMS)

method, [1478](#)

Composite shell elements, [203](#)

specification of layer thicknesses,
[204](#)

transverse shear stresses, [203](#)

- Computers supported by ADINA, 2
- Concentrated dampers, 1195
- Concentrated loads, 1050
- Concrete material model, 823
 - compressive failure envelopes, 602
 - data fitted, 588, 607
 - material behavior after failure, 603
 - material failure envelopes, 595
 - Poisson's ratio in the compressive region, 606
 - post tensile cracking behavior, 603
 - postcompression crushing behavior, 605
 - stress-strain relations, 591
 - temperature effects, 606
 - tensile failure envelopes, 599
- Concrete material models, 588
- Condensed node, 1386
- Connector elements, 393
- Consistent contact surface stiffness, 966
- Consolidation analysis, 733
- Constraint equations, 1101
- Contact bodies, 907
- Contact group birth/death, 895
- Contact oscillations, suppressing, 929
- Contact pairs, 913
- Contact slip, 1093
- Contact surface depth, 910
- Contact surface offsets, 907
- Contact surfaces
 - forces, 980
 - three-dimensional, 897
 - two-dimensional, 896
- Contact surfaces extension, 905, 1119
- Contact variables, 1508, 1538
- Convergence criteria, 1170
 - contact, 1172
 - energy, 1171, 1174, 1179
 - force and moment, 1171, 1174, 1179
 - translation and rotation, 1172, 1179
- Coordinate variables, 1509, 1535
- Crack propagation, 1271, 1313
 - crack growth control, 1315
 - definition of the crack propagation surface, 1314
 - modeling of the crack advance, 1318, 1320, 1321
 - node release-only technique, 1313
 - pressure loading, 1321
- Crack resistance curves, 1316
- Crack tip blunting, 1302
- CRACK-M meshing features, 1327
- Creep coefficients
 - user-supplied, 558, 580
- Creep laws, 555, 577
 - eight-parameter, 555, 578
 - eight-parameter with variable coefficients, 557, 580
 - exponential, 555, 578
 - LUBBY2, 556, 578
 - LUBBY2 with variable coefficients, 558, 581
 - power, 555, 578
- Creep material model, 11, 543, 555, 577
- Creep strains, 555, 577
 - O.R.N.L. rules for cyclic loading conditions, 559, 581
 - strain hardening, 559, 581
- Creep-variable material model, 543, 557, 580
- Curve description material model, 705
 - cracking, 709
 - cyclic parts, 1401
 - tension cut-off, 709
- Curve fitting for rubber and foam material models, 663
 - Arruda-Boyce model, 667
 - eight-chain model, 669
 - hyper-foam model, 667

Mooney-Rivlin model, 665
Ogden model, 666
Cyclic symmetry analysis, 1397
periodic symmetry, 1399
restrictions, 1402

D

Damping, 1195, 1197
consistent, 1196
lumped, 1195
modal, 1194
Rayleigh, 1196
Strain Energy Proportional, 1197
Darcy's law, 733
Data fitted concrete material model, 588, 607
Deformation gradient tensor
elastic, 465
inelastic, 465
total, 461
Deformation-dependent distributed loads, 1055
Deformation-dependent pressure loads, 1055
Determinant search method, 1137
Direct time integration, 341
Director vectors, 181
Disk files, 1444
displacements, 1447
forces, 1447
pipe internal pressures, 1446
temperature gradients, 1447
temperatures, 1446
Displacement variables, 1511
Displacement-based finite elements, 19, 42
Displacement-based fluid elements, 284
3-D, 285
axisymmetric, 285
element output, 286

irrotational conditions, 284
planar, 285
Distributed loads, 1054
beam, 1059
deformation dependent, 1055
iso-beam, 1060
pipe, 1060
shell, 1061
Draw bead variables, 1508
Drucker-Prager material model, 711

E

Earthquake loading, 1194
Effective plastic strain, 493
Effective-stress-function algorithm, 511
Eigenvalue problems, 1127
Eigenvector variables, 1533
eight-chain material model
3-D analysis, 662
axisymmetric analysis, 662
plane strain analysis, 662
plane stress analysis, 662
selection of material constants, 662
Eight-chain material model, 653
Eight-parameter creep law
variable coefficients, 557, 580
Eight-parameter creep law, 555
Eight-parameter creep law, 578
Elastic-isotropic material model, 471, 472
Elastic-orthotropic material model, 471, 472
2-D solid elements, 472
3-D solid elements, 477
fabric, 728
plate elements, 477
shell elements, 478
Electric field variables, 1528
Electromagnetic loads, 1091
Element 'death upon rupture', 1412

- Element birth/death, 1153, 1404
 - Element local node field variables, 1509
 - Element local node variables, 1509
 - Element locking, 226
 - Element output for
 - 2-D solid elements, 29
 - 3-D solid elements, 53
 - alignment elements, 385
 - axisymmetric shell elements, 175
 - cohesive elements, 392
 - displacement-based fluid elements, 284
 - general elements, 266
 - iso-beam elements, 176
 - nonlinear spring/damper/mass elements, 273
 - pipe elements, 261
 - potential-based fluid elements, 342
 - shell elements, 217
 - truss elements, 13
 - Element/layer field variables, 1508
 - Elements
 - 2-D solid, 17
 - 3-D solid, 40
 - 3D-shell, 232
 - alignment, 343
 - axisymmetric shell, 165
 - cable, 9
 - cohesive, 388
 - connector, 393
 - general, 264
 - iso-beam, 165
 - pipe, 244
 - plate, 179
 - RBE3, 1110
 - rebar, 14
 - ring, 9
 - shell, 180
 - spring/damper/mass, 264
 - End-release option, 155
 - Energy release rate, 1274
 - Energy variables, 1539
 - Engineering strains, 455, 458
 - Engineering stresses, 455, 459
 - Enriched Bathe subspace iteration method, 1139
 - Envelope response type, 1561
 - Equilibrium iterations
 - BFGS method, 1155
 - full Newton method, 1155
 - modified Newton method, 1154
 - Error indicators, 1565
 - Explicit dynamic analysis, 1215
 - Exponential creep law, 555, 578
- F**
- Fabric material model, 728
 - Failure criteria, 209
 - Hashin failure criterion, 215
 - maximum strain failure criterion, 211
 - maximum stress failure criterion, 209
 - tensor polynomial failure criterion, 213
 - Tsai-Hill failure criterion, 212
 - user-supplied failure criterion, 216
 - Failure criterion variables, 1530
 - Fiber-matrix composites, 204
 - Five degrees of freedom node, 191
 - Flanges, 248, 251
 - Floor response spectrum analysis, 1258
 - curve smoothing, 1262
 - damped transfer function method, 1268
 - peak broadening, 1262
 - response spectra loading, 1263
 - spatial combinations, 1268
 - time history loading, 1260
 - undamped transfer function, no

- resonance correction method,
1267
- undamped transfer function,
resonance correction method,
1267
- Fluid variables, 1514
- Follower loads, 1053
- Force variables, 1522
- Formulations for
 - 2-D solid elements, 24
 - 3-D solid elements, 49
 - axisymmetric shell elements, 174
 - iso-beam elements, 174
 - pipe elements, 254
 - shell elements, 188
 - truss elements, 11
- Fourier analysis, 1242
 - power spectral density, 1243
- Fracture mechanics, 1271
 - crack propagation, 1271, 1313
 - line contour method, 1271
 - meshing recommendations, 1324
 - virtual crack extension method,
1271
- Free surface modeling, 288
- Frequency analysis, 1133
 - Bathe subspace iteration method,
1137
 - error bound, 1136
 - Lanczos iteration method, 1140
 - potential-based fluid elements, 306,
341
 - restart, 1441
 - rigid body modes, 1136
- Frequency domain analysis, 1231
 - Fourier analysis, 1231
 - harmonic vibration analysis, 1231
 - random vibration analysis, 1231
 - response spectrum analysis, 1231
 - SDOF system response, 1231
- Frequency/mode variables, 1538

- Friction
 - user-supplied models, 867
- Friction delay, 965
- Full Newton iterations, 1153, 1155,
1176
- line searches, 1156

G

- Gap conductance, 889
- Gasket material model, 738
- General elements, 264, 1195
 - element output, 266
- Geometric imperfections, 1132, 1454
- Geotechnical material models, 705
- Global mass matrix, 1205
- Green-Lagrange strains, 456, 458
- Gurson material model, 488

H

- Harmonic response types, 1560
- Harmonic vibration analysis
 - applied forces, 1245
 - ground motion direction, 1252
 - ground motions, 1246
 - initial calculations, 1251
 - modal damping ratios, 1252
 - phase angle, 1252
 - potential-based fluid elements, 309,
341
 - quasi-static amplitude, 1249
 - real part of the response, 1252
 - RMS amplitude, 1252
 - signed amplitude, 1252
 - sweep spectra, 1252
- Hencky strain, 458
- Hermitian beam elements, 63
- Hill yield condition, 508
- Holzapfel model for finite strain
 - viscoelasticity, 673
- Hydro-pressure, 1121

Hydrostatic fluid loading, 1121
 Hyper-foam material model, 643
 3-D analysis, 644
 axisymmetric analysis, 644
 plane strain analysis, 644
 plane stress analysis, 644
 selection of material constants, 644

I

Improperly supported bodies, 996
 Incompatible modes finite elements, 22, 46
 Inelastic deformations, 464
 Inertial properties, 1437
 center of mass, 1437
 centroid, 1437
 element group, 1437
 moments of inertia, 1437
 products of inertia, 1437
 total mass, 1437
 total volume, 1437
 Infinite elements, 288
 Initial accelerations, 1413
 Initial conditions, 1413
 initial accelerations, 1413
 initial displacements, 1413
 initial pipe internal pressures, 1415
 initial strains, 1415
 initial stresses, 1423
 initial temperatures, 1414
 temperature gradients, 1414
 Initial displacements, 1413
 Initial loads, 1095
 Initial pipe internal pressures, 1415
 Initial strains, 1415
 element, 1417
 large strain formulations, 1422
 nodal point, 1417
 strain fields, 1419, 1420
 Initial stresses, 1423
 restrictions, 1426

Initial temperature gradients, 1414
 Initial temperatures, 1414
 Initial thermal strains, 1415
 Initial velocities, 1413
 Initial conditions
 initial velocities, 1413
 Irradiation creep material model, 544, 558, 581
 Iso-beam elements, 165
 element output, 174, 176
 formulations, 174
 general 3-D, 165
 linear, 171
 mass matrices, 176
 material models, 174
 nonlinear, 174
 numerical integration, 169, 175
 plane strain 2-D, 165
 plane stress 2-D, 165
 shear deformations, 169
 warping effects, 168
 Isothermal plasticity material models, 488
 Isotropic hardening, 489, 516
 Isotropic hyperelastic effects, 634
 Iterative solvers, 1148

J

Jaumann strain, 458
 J-integral, 1271

K

Kinematic hardening, 489, 516
 Kirchhoff stresses, 460

L

Lanczos iteration method, 1140
 Large displacement formulation, 11, 174, 254, 471, 541, 588, 705
 Large displacement/large strain

- formulation, [25](#), [49](#), [456](#), [457](#), [490](#),
[508](#), [515](#), [545](#), [633](#), [711](#), [730](#), [733](#)
 - Large displacement/small strain
 - formulation, [25](#), [49](#), [456](#), [490](#), [508](#),
[515](#), [517](#), [545](#), [566](#), [711](#), [728](#), [730](#),
[733](#)
 - Large strain analysis
 - ULJ formulation, [466](#)
 - Latent heat variables, [1514](#)
 - LDC method, [1047](#), [1167](#), [1176](#)
 - constant increment of external work
method, [1168](#)
 - constant spherical arc length
constraint method, [1168](#)
 - load vector multiplier, [1169](#)
 - Limiting maximum incremental
displacement, [999](#)
 - Line contour method, [1271](#), [1279](#)
 - output, [1307](#)
 - Line contour variables, [1509](#)
 - Line search, [1157](#)
 - Linear buckling analysis
 - secant formulation, [1129](#)
 - Linear formulation, [11](#), [25](#), [49](#), [191](#),
[254](#), [471](#)
 - Linear static analysis, [1145](#), [1225](#),
[1226](#)
 - Linearized buckling analysis
 - classical formulation, [1128](#)
 - error bound, [1131](#)
 - restart analysis, [1129](#)
 - Load penetration, [1057](#)
 - Load step response type, [1559](#)
 - Load step results, [1559](#)
 - Load/displacement incrementation,
[1153](#)
 - Loading
 - centrifugal, [1071](#)
 - concentrated, [1050](#)
 - distributed, [1054](#)
 - Electromagnetic, [1091](#)
 - hydrostatic fluid, [1121](#)
 - initial, [1095](#)
 - mass-proportional, [1077](#)
 - pipe internal pressures, [1090](#)
 - prescribed temperature gradients,
[1087](#)
 - prescribed temperatures, [1087](#)
 - pressure, [1053](#)
 - user-supplied, [1095](#)
 - Location-independent variables, [1508](#)
 - Logarithmic strains, [458](#)
 - Low speed dynamics, [1158](#)
 - LUBBY2 creep law, [556](#), [578](#)
 - variable coefficients, [558](#), [581](#)
- ## M
- Mapping files, [1087](#), [1447](#)
 - Mass
 - inertial properties, [1437](#)
 - Mass matrices for
 - 2-D elements, [28](#)
 - iso-beam elements, [176](#)
 - pipe elements, [260](#)
 - shell elements, [208](#), [209](#)
 - truss elements, [12](#)
 - Mass matrix, [1222](#)
 - Mass scaling, [1228](#)
 - Mass-proportional loads, [1077](#)
 - potential-based fluid elements, [322](#)
 - Material models, [455](#)
 - Anand, [756](#)
 - Arruda-Boyce, [641](#)
 - Cam-clay, [716](#)
 - cast iron, [784](#)
 - concrete, [823](#)
 - concrete, [588](#)
 - creep, [543](#)
 - creep variable, [557](#), [580](#)
 - creep-variable, [543](#)
 - curve description, [705](#)
 - curve fitting, [663](#)

- Drucker-Prager, 711
- eight-chain, 653
- elastic-isotropic, 471, 472
- elastic-orthotropic, 471, 472
- fabric, 728
- gasket, 738
- geotechnical, 705
- Gurson, 514
- hyper-foam, 643
- irradiation creep, 544, 558, 581
- isothermal plasticity, 488
- isotropic hyperelastic effects, 634
- Mohr-Coulomb, 720
- Mooney-Rivlin, 636
- Mroz-bilinear, 500
- Mullins effect, 687
- multilinear-plastic-creep, 543, 555, 577
- multilinear-plastic-creep-variable, 543, 557, 580
- nonlinear elastic, 481, 485
- Ogden, 639
- orthotropic effect, 691
- piezoelectric, 759
- plastic, 821
- plastic creep, 543, 555, 577
- plastic-bilinear, 489
- plastic-creep-variable, 543, 557, 580
- plastic-cyclic material, 516
- plastic-multilinear material, 489
- plastic-orthotropic, 508
- Ramberg-Osgood, 828
- rubber and foam, 633
- Shape Memory Alloy, 747
- SMA, 747
- Sussman-Bathe, 645
- temperature dependence, 697
- thermal strain effect, 695
- thermo-cast iron, 784
- thermo-elastic, 541
- thermoelasticity and creep, 821
- thermo-elasto-plasticity and creep, 543
- thermo-isotropic, 541
- thermo-orthotropic, 541
- thermo-plastic, 543
- thermo-plastic-cyclic, 565
- user-coded, 796
- viscoelastic, 729, 827
- viscoelastic effects, 672
- viscoplastic, 814
- Material models for
 - 2-D solid elements, 24
 - 3-D solid elements, 48
 - axisymmetric shell elements, 174
 - iso-beam elements, 174
 - pipe elements, 253
 - plate elements, 179
 - shell elements, 188
 - truss elements, 11
- Materially-nonlinear-only
 - formulation, 11, 25, 49, 174, 191, 254, 491, 508, 515, 535, 541, 545, 566, 588, 705, 711, 716, 730, 733
- Matrices for
 - 3-D solid elements, 53
 - axisymmetric shell elements, 176
- Mean-square response, 1252
 - interpretation of, 1256
- Memory allocation, 1146
 - in-core solution, 1444
 - out-of-core solution, 1444
- Mesh evaluation, 1565
- Mesh glueing, 1115
- Mesh surface variables, 1509
- Miscellaneous variables, 1539
- Mixed-interpolated finite elements, 21, 44
- Mixed-interpolation formulation, 25, 49, 492, 536, 545, 566, 573, 637, 734, 797

Modal damping, [1194](#)
Modal mass, [1233](#)
Modal participation factors
 properties of, [1239](#)
Modal stresses, [1141](#)
Mode shape response type, [1560](#)
Mode shape results, [1559](#)
Mode superposition, [1194](#), [1204](#)
 potential-based fluid elements, [307](#)
Modeling of gaps, [486](#), [500](#)
Modified Newton iterations, [1153](#),
 [1154](#)
 line searches, [1154](#)
Mohr-Coulomb material model, [720](#)
Moment-curvature beam elements,
 [144](#)
 axial behavior, [145](#)
 cyclic behavior, [152](#)
 elastic-plastic material model, [147](#)
 element output, [153](#)
 flexural behavior, [145](#)
 mass matrices, [153](#)
 nonlinear elastic model, [146](#)
 numerical integration, [146](#)
 section results, [153](#)
 stress resultants, [153](#)
 torsional behavior, [145](#)
Moments of inertia
 inertial properties, [1437](#)
Mooney-Rivlin material model, [636](#)
 3-D analysis, [637](#)
 axisymmetric analysis, [637](#)
 plane strain analysis, [637](#)
 plane stress analysis, [637](#)
 selection of material constants, [638](#)
Mroz-bilinear material model, [500](#)
Mullins effect, [687](#)
Multigrid solver, [1150](#)
Multilayer shell elements, [203](#)
 specification of layer thicknesses,
 [204](#)

 transverse shear stresses, [203](#)
Multilinear-plastic-creep material
 model, [543](#), [555](#), [577](#)
Multilinear-plastic-creep-variable
 material model, [543](#), [557](#), [580](#)

N

Newmark method, [1194](#)
Node discrete variables, [1509](#)
Node field variables, [1508](#)
Node release-only technique, [1313](#)
Node shift/release technique, [1313](#)
Node-node contact, [912](#)
Noh-Bathe method, [1220](#), [1223](#)
Nominal strains, [459](#)
Nonconvergence, [1173](#), [1175](#)
Nonlinear elastic material model, [481](#),
 [485](#)
Nonlinear static analysis, [1152](#)
 selection of incremental solution
 method, [1175](#)
Non-positive definite stiffness matrix,
 [1147](#)
Numerical integration for
 2-D solid elements, [26](#)
 3-D solid elements, [50](#)
 axisymmetric shell elements, [169](#),
 [175](#)
 iso-beam elements, [169](#), [175](#)
 pipe elements, [257](#)
 shell elements, [200](#)
 truss elements, [11](#)
NVS virtual shifts, [1271](#), [1338](#)

O

O.R.N.L. rules for cyclic loading
 conditions, [559](#), [581](#)
Ogden material model, [639](#)
 3-D analysis, [640](#)
 axisymmetric analysis, [640](#)

plane strain analysis, 640
 plane stress analysis, 639
 selection of material constants, 640
 Operating systems supported by
 ADINA, 2
 Ovalization degrees of freedom, 245,
 251

P

Parallel processing
 sparse solver, 1443
 Parallel-Network Framework, 769
 Periodic symmetry, 1399
 Piezoelectric analysis, 1496
 Piezoelectric material model, 759
 Pipe elements, 244
 element output, 261
 formulations, 254
 mass matrices, 260
 material models, 253
 numerical integration, 257
 ovalization degrees of freedom, 245
 pipe-beam, 244
 pipe-shell, 244
 recommendations for use, 263
 warping degrees of freedom, 245
 Pipe internal pressures, 248, 254,
 1090
 Plastic strains, 547, 565
 Plastic-bilinear material model, 489
 Plastic-creep material model, 543,
 555, 577
 Plastic-creep-variable material model,
 543, 557, 580
 Plastic-cyclic material model, 516
 Plastic-multilinear material model,
 489
 Plastic-orthotropic material model,
 508
 Plate elements, 179
 material models, 179

Pore pressures, 733
 Porous media formulation, 733
 Porous structures, 733
 Positive definite stiffness matrix, 1145
 Post-collapse response, 1167, 1177
 Postprocessing, 1501
 Potential-based fluid elements, 287
 3-D, 287
 axisymmetric, 287
 choice of formulation, 311
 direct time integration, 341
 element output, 342
 free surfaces, 288
 ground motions, 302
 infinite elements, 288
 initial pressure in, 339
 mass-proportional loads, 322
 planar, 287
 pressure loads, 322
 static analysis, 340
 Power creep law, 555, 578
 Power spectral density, 1243, 1253
 Prescribed accelerations, 1080
 Prescribed displacements, 1080
 Prescribed load variables, 1534
 Prescribed temperature gradients,
 1087
 Prescribed temperatures, 1087
 Prescribed velocities, 1080
 Pressure loads, 1053
 deformation-dependent, 1055
 load penetration, 1057
 plate/shell, 1061
 potential-based fluid elements, 322
 shell, 1061
 three-dimensional, 1056
 two-dimensional, 1056
 Products of inertia
 inertial properties, 1437

Q

Quarter-point midside nodes, [1300](#),
[1325](#)

Quasi-static amplitude, [1249](#)

R

Radiation field variables, [1509](#)

Ramberg-Osgood material model, [828](#)

Random response type, [1258](#), [1560](#)

Random response types, [1252](#)

Random vibration analysis, [1252](#)

 ground motion direction, [1257](#)

 initial calculations, [1257](#)

 mean-square response, [1252](#)

 modal damping ratios, [1257](#)

 potential-based fluid elements, [309](#),
 [341](#)

 power spectral density, [1253](#), [1257](#)

Rayleigh damping, [1196](#), [1223](#)

RBE3 element, [1110](#)

Reaction variables, [1526](#)

Reactions, [1403](#)

Rebar elements, [14](#)

Recommendations for use of
 shell elements, [224](#)

Remeshing, [1447](#)

Residual response type, [1560](#)

Residual results, [1559](#)

Response spectrum analysis, [1231](#)

 absolute sum method, [1234](#)

 algebraic sum method, [1235](#)

 CQC method, [1236](#)

 double sum method, [1235](#)

 DSC method, [1235](#)

 ground motion direction, [1241](#)

 initial calculations, [1239](#)

 loading response spectra, [1241](#)

 modal combinations, [1234](#)

 modal damping ratios, [1241](#)

 modal-combinations, [1241](#)

 potential-based fluid elements, [309](#),
 [341](#)

 residual responses, [1232](#)

 residual terms, [1236](#), [1241](#)

 spatial combinations, [1241](#)

 SRSS method, [1234](#)

 static correction terms, [1236](#)

 static load case, [1239](#)

 ten percent method, [1234](#)

Response spectrum response type,
[1560](#)

Response spectrum analysis

 static load case, [1242](#)

Response type

 response-combination, [1241](#)

Response types

 envelope, [1561](#)

 harmonic, [1560](#)

 load step, [1559](#)

 mode shape, [1560](#)

 random, [1252](#), [1258](#), [1560](#)

 residual, [1560](#)

 response spectrum, [1560](#)

 response-spectrum, [1241](#)

Response-combination response type,
[1241](#), [1560](#)

Response-spectrum response type,
[1241](#)

Restart, [1438](#)

Restart option

 frequency analysis, [1441](#)

Result calculation within the AUI,
[1501](#)

 combination transformations, [1502](#)

 examples, [1502](#)

 resultant transformations, [1502](#)

 resultants after combinations, [1504](#)
 resultants before combinations,
 [1504](#)

 spatial transformations, [1502](#)

 time transformations, [1502](#)

-
- Result locations
 - contact segments, 1549
 - draw bead segments, 1550
 - element section, 1549
 - element/layer, 1544
 - line contours, 1551
 - location- independent, 1544
 - mesh integration points, 1552
 - node, 1544
 - radiosity segments, 1550
 - virtual shifts, 1550
 - Result recovery, 1438, 1441
 - Results
 - load step, 1559
 - mode shape, 1559
 - residual, 1559
 - Resuses (of substructures), 1386
 - Retained nodes, 1386
 - Rho_∞-Bathe method, 1190
 - Rigid body modes, 1136
 - Rigid links, 1103
 - Rigid target contact algorithm, 1003
 - Ring element, 9
 - Rubber and foam material models, 633
 - Rubber stability indicators, 699
 - Rupture conditions, 498, 514, 536, 546, 554, 561, 574, 583
- S**
- SDOF system response, 1258
 - Section field variables, 1509
 - Shape Memory Alloy, 747
 - Shape Memory Alloy material model, 747
 - Shape memory effect, 747
 - Shell elements, 180
 - 16-node, 226
 - 3D, 232
 - 4-node, 224
 - 9-node, 226
 - basic assumptions in, 180
 - composite, 203
 - director vectors, 181
 - displacement-dependent pressure loading, 191
 - element output, 217
 - failure criteria, 209
 - locking, 226
 - mass matrices, 208, 209
 - material models, 188
 - MITC16, 226
 - MITC4, 224
 - MITC9, 226
 - multilayer, 203
 - nodal forces, 223
 - nodal point degrees of freedom, 191
 - numerical integration, 200
 - recommendations for use, 224
 - section results, 220
 - shear deformations, 186
 - stress resultants, 220
 - thick, 224
 - thin, 224
 - transition, 198
 - Shell midsurfaces
 - plotting results onto, 1559
 - Shell stiffeners, 169
 - Shell-shell intersections, 199
 - Shell-solid intersections, 199
 - Shifting procedure, 1141
 - Six degrees of freedom node, 193
 - Skew systems, 1045
 - SMA, 747
 - SMA material model, 747
 - Small displacement contact feature, 930
 - Small displacement formulation, 11, 174, 254, 471, 541, 588, 705, 716
 - Small displacement/small strain formulation, 25, 49, 455, 490, 508, 515, 517, 545, 566, 711, 730, 733
-

- Smoothing, 1557
 - restrictions, 1558
 - Soil consolidation analysis, 1493
 - Solution graphs, 1211
 - Solution of $KU=R$, 1145
 - iterative solvers, 1148
 - multigrid solver, 1150
 - sparse solver, 1146
 - Solvers, 1209, 1229
 - 3D-iterative solver, 1150
 - Sparse matrix solver, 1146
 - Sparse solver, 1443
 - in-core, 1147
 - memory allocation, 1146
 - out-of-core, 1147
 - parallel processing, 1443
 - Spring/damper/mass elements, 264, 1195
 - linear single-degree-of-freedom, 267
 - linear two-degree-of-freedom, 268
 - nonlinear, 269
 - Stabilized TLA method, 1164
 - Static analysis, 1145, 1225, 1226
 - potential-based fluid elements, 303, 340
 - Stiffness stabilization, 997, 1153
 - Strain energy density, 1435
 - Strain Energy Proportional Damping, 1197
 - Strain hardening, 559, 581
 - Strain measures, 455, 458
 - engineering strains, 455, 458
 - Green-Lagrange strains, 458
 - Hencky strain, 458
 - Jaumann strain, 458
 - logarithmic strains, 458
 - stretches, 458
 - Strain variables, 1518
 - Strain-rate effects
 - 2-D elements, 499
 - 3-D elements, 499
 - shell elements, 499
 - Stress intensity factors, 1271, 1275
 - Stress measures, 455, 459
 - 2nd Piola-Kirchhoff stress, 459
 - Cauchy stress, 459
 - Cauchy stresses, 455
 - engineering stress, 459
 - engineering stresses, 455
 - Kirchhoff stress, 460
 - Stress variables, 1515
 - Stretch tensor
 - left, 464
 - Stretches, 458
 - Sturm sequence check, 1139, 1141
 - Subspace iteration method
 - Sturm sequence check, 1139, 1141
 - Substructures, 1386
 - Substructuring, 1385
 - condensed nodes, 1386
 - linear analysis, 1385
 - nonlinear analysis, 1387
 - restrictions, 1388
 - retained nodes, 1386
 - Superelastic effect, 747
 - Supported computers and operating systems, 2
 - Suppressing contact oscillations, 929
 - Sussman-Bathe material model, 645
 - SVS virtual shifts, 1271, 1356
 - Swelling analysis, 733
 - Symmetry condition (for pipe elements), 252
- T**
- Temperature files, 1087
 - Temperature variables, 1514
 - Temperature-dependent material, 697
 - Thermal coupling between solids and fluids, 1498
 - Thermal strains, 468, 546

Thermo-cast iron material model, 784
Thermo-elastic material models, 541
Thermoelasticity and creep material model, 821
Thermo-elasto-plasticity and creep material models, 543
Thermo-isotropic material model, 541
Thermo-mechanical coupling
 heat transfer between contact surfaces, 887
 surface heat generation due to frictional contact, 892
Thermo-orthotropic material model, 541
Thermo-plastic material model, 543
Thermo-plastic-cyclic material model, 565
Thermorheologically simple material, 633, 697
Thickness variables, 1537
Tied contact, 933
Time functions, 1046, 1048
 constant, 1048
 short circuit, 1048
 sinusoidal, 1048
Time step management, 1227
Time variables, 1537
TL formulation, 26, 49, 471, 491, 508, 535, 541, 545, 566, 588, 633, 705, 711, 716, 730, 733, 796
TLA method, 1164
TLA-S method, 1164
Total Load Application method, 1164
Traction variables, 1524
TRS material, 633, 697
True strains, 459
Truss elements, 9
 element output, 13
 formulations, 11
 mass matrices, 12
 material models, 11

numerical integration, 11
rebar, 14
recommendations for use, 14

U

UL formulation, 471, 491, 541, 545
ULH formulation, 49, 456, 457, 458, 535, 545, 566, 730, 734, 797
ULJ formulation, 456, 457, 458, 491, 508, 535, 566
Undrained analysis, 733
User-coded material model, 796
User-defined variables, 1542
User-supplied creep coefficients, 558, 580
User-supplied elements, 277
User-supplied friction models, 867
User-supplied loads, 1095

V

Variables, 1508
 acceleration, 1513
 contact, 1508
 coordinate, 1509, 1535
 displacement, 1511
 draw bead, 1508
 eigenvector, 1533
 electric field, 1528
 element local node, 1509
 element local node field, 1509
 element/layer field, 1508
 failure criterion, 1530
 fluid, 1514
 force, 1522
 frequency/mode, 1538
 interpolation within elements, 1555
 latent heat, 1514
 line contour, 1509
 location-independent, 1508
 mesh surface, 1509

- miscellaneous, [1538](#), [1539](#)
 - node discrete, [1509](#)
 - node field, [1508](#)
 - prescribed load, [1533](#), [1534](#)
 - radiation field, [1509](#)
 - reaction, [1526](#)
 - section field, [1509](#)
 - smoothing, [1557](#)
 - strain, [1518](#)
 - stress, [1515](#)
 - temperature, [1514](#)
 - thickness, [1537](#)
 - time, [1537](#)
 - traction, [1524](#)
 - User-defined, [1542](#)
 - velocity, [1512](#)
 - virtual shift, [1509](#)
 - Velocity variables, [1512](#)
 - Virtual crack extension method, [1271](#),
[1281](#)
 - description of virtual shifts, [1308](#)
 - dynamic correction, [1286](#)
 - output, [1313](#), [1355](#), [1380](#)
 - pressure correction, [1285](#)
 - thermal correction, [1285](#)
 - Virtual shift variables, [1509](#)
 - Virtual shifts, [1308](#)
 - spatially fixed, [1315](#)
 - spatially moving, [1315](#)
 - Viscoelastic effects, [672](#)
 - Viscoelastic material model, [729](#), [827](#)
 - Prony series, [732](#)
 - Viscoplastic material model, [814](#)
 - Volume
 - inertial properties, [1437](#)
 - von Karman ovalization modes, [249](#)
- W**
- Wagner effect, [79](#)
 - Warping degrees of freedom, [245](#), [251](#)
- Z**
- Zero-slope-of-pipe-skin, [251](#)