AUTOMATIC

DYNAMIC

INCREMENTAL

NONLINEAR

ANALYSIS

Theory and Modeling Guide

Volume III: CFD & FSI

ADINA 9.6 April 2020

ADINA R & D, Inc.

ADINA
Theory and Modeling Guide

Volume III:
CFD & FSI

April 2020

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

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

April 2020 Printing
Printed in the USA

Table of contents

Nomenclature **11**

Chapter 1 Introduction **24**

Chapter 2 Various forms of fluid equations **28**
2.1 General conservative Navier-Stokes equations 28
2.2 Nonconservative Navier-Stokes equations 30
2.3 Rotating frame of reference 31
2.4 Planar flow 34
2.5 Axisymmetric flow 35
2.6 Incompressible flow 37
2.7 Fluid flow through porous media 42
2.8 Slightly compressible flow 46
2.9 Compressible flows 48
2.10 High-speed compressible flow 49
2.11 Turbulence models 51
2.12 Mass transfer 64
2.13 Steady-state equations 70
2.14 Arbitrary-Lagrangian-Eulerian formulations 70
2.15 System units and nondimensional forms 74

Chapter 3 Formulation of incompressible, slightly compressible and low-speed compressible flows **79**
3.1 Governing equations 82
3.2 Numerical method 82
3.2.1 Time integration 82
3.2.2 Discretized equations 84
3.2.3 Upwinding techniques 85
3.2.4 Solution method for nonlinear finite element equations 87
3.3 Elements 89
3.4 Boundary conditions 90
3.4.1 General descriptions 90

3.4.2 Usual boundary conditions for fluid 93
3.4.3 Special boundary conditions for fluid 98
3.4.4 Usual boundary conditions for temperature 122
3.4.5 Special boundary conditions for temperature 124
3.4.6 Thermal resistance condition 130
3.4.7 User-supplied boundary condition 130
3.4.8 Shell-thermal condition 138
3.4.9 Mass-flow-rate condition 139
3.5 Initial conditions 140
3.6 Material models 141
3.6.1 Constant material model 141
3.6.2 Time-dependent model 142
3.6.3 Power-law model 142
3.6.4 Carreau model 143
3.6.5 Temperature-dependent model 143
3.6.6 Temperature-dependent power-law model 144
3.6.7 Second order model 144
3.6.8 Large-eddy-simulation model 145
3.6.9 User-supplied materials 146
3.6.10 ASME steam table 153
3.6.11 P-T-dependent material 154
3.6.12 Field friction 154
3.6.13 User-modified materials 155
3.6.14 Material curves 155
3.6.15 Gas models 157
3.6.16 Conditional loads 161

Chapter 4 Formulation of fluid flows through porous media 167
4.1 Governing equations 168
4.2 Numerical method 169
4.3 Elements 170
4.4 Boundary conditions 170
4.5 Initial conditions 172
4.6 Material model 172

Chapter 5 Formulation of turbulence in incompressible, slightly compressible and low-speed compressible flows 175
5.1 Governing equations 177

5.2 Numerical method 177
5.3 Elements 179
5.4 Boundary conditions 179
5.4.1 Prescribed turbulence K-ε 180
5.4.2 Prescribed turbulence K-ω 181
5.4.3 Zero flux of turbulence variables 182
5.4.4 Solid walls 182
5.4.5 User-supplied boundary condition 184
5.5 Initial conditions 185
5.6 Material models 185
5.6.1 K-ε turbulence model 185
5.6.2 RNG K-ε turbulence model 187
5.6.3 High Reynolds K-ω turbulence model 187
5.6.4 Low Reynolds K-ω turbulence model 187
5.6.5 Shear Stress Transport model 188
5.6.6 Spalart-Allmaras and Detached Eddy Simulation material models 188
5.6.7 Two-layer zonal material model 189
5.6.8 Material curves 189
5.6.9 Gas models 189
5.6.10 K-ε realizable turbulence model 189

Chapter 6 Formulation of high-speed compressible flows 191
6.1 Governing equations 193
6.2 Numerical method 193
6.2.1 Geometric conservation law in moving meshes 194
6.2.2 Flux-Splitting method 196
6.2.3 Finite element method 198
6.2.4 Solution procedure 199
6.2.5 Explicit time integration scheme 199
6.3 Control volumes and finite elements 201
6.3.1 Control volume in triangular elements (3-node) 201
6.3.2 Control volume in tetrahedral elements (4-node) 203
6.4 Boundary conditions 204
6.4.1 General descriptions 204
6.4.2 Usual boundary conditions for fluid 212
6.4.3 Special boundary conditions 216
6.4.4 User-supplied boundary condition 229

6.5 Initial conditions....230
6.6 Material models....230
6.6.1 Constant material model....230
6.6.2 Power-law model....231
6.6.3 Temperature-dependent model....231
6.6.4 Pressure-dependent model....232
6.6.5 Pressure-temperature-dependent model....232
6.6.6 User-supplied materials....232

Chapter 7 Formulation of turbulence in high-speed compressible flows....235
7.1 Governing equations....236
7.2 Numerical method....237
7.3 Elements....238
7.4 Boundary conditions....238
7.4.1 Prescribed turbulence K-ε....238
7.4.2 Zero flux of turbulence variables....239
7.4.3 Solid walls....239
7.4.4 User-supplied boundary condition....241
7.5 Initial conditions....241
7.6 Material models....241

Chapter 8 Formulation of mass transfer in fluid flows....243
8.1 Governing equations....244
8.2 Numerical method....244
8.3 Elements....246
8.4 Boundary conditions....246
8.4.1 Prescribed mass-ratio....246
8.4.2 Distributed mass flux....247
8.4.3 Mass Convection....247
8.4.4 User-supplied boundary condition....248
8.5 Initial conditions....248
8.6 Material models....249
8.6.1 Constant material data model....249
8.6.2 Velocity-dependent model....249
8.6.3 Pressure-temperature-dependent model....250
8.6.4 User-supplied materials....250

Chapter 9 Formulation of fluid-structure interactions 257
9.1 Introduction 257
9.2 Theory 258
9.2.1 Kinematic and dynamic conditions 259
9.2.2 Separate meshes of fluid and solid models 260
9.2.3 Consistent time integration for fluid and solid models 263
9.2.4 Finite element equations of the coupled system 264
9.3 Iterative computing of two-way coupling 266
9.4 Direct computing of two-way coupling 267
9.5 Direct computing of one-way coupling 269
9.6 Indirect computing of one-way coupling 270
9.7 Elements on fluid and solid interfaces 271
9.8 Material models 278
9.9 Procedure of model preparation and testing 279
9.10 Other types of fluid-structure interactions 281
9.10.1 Fluid-solid interaction in porous media 283
9.10.2 Coupled thermal fluid-structural interaction 284
9.10.3 Model one-way thermally coupled problem using mapping 285

Chapter 10 Elements 287
10.1 Boundary line elements 287
10.2 2D triangular element (3-node) 288
10.3 2D quadrilateral element (9-node) 290
10.4 2D triangular element (6-node) 291
10.5 3D tetrahedral element (4-node) 293
10.6 3D brick element (27-node) 294
10.7 2D FCBI elements (4- and 3-node) 296
10.8 3D FCBI elements (8-, 4-, 6- and 5-node) 298
10.9 FCBI-C elements 300

Chapter 11 Solutions of nonlinear equations 303
11.1 Newton-Raphson method 303
11.2 Segregated method 304
11.2.1 Basic parameters 306
11.2.2 Outer iteration control 308

11.2.3 Inner iteration control....314
11.2.4 Comments and tips on convergence....316
11.2.5 Limitations....320
11.3 Solvers of algebraic equations....321
11.3.1 Gauss elimination method....321
11.3.2 Sparse solver....322
11.3.3 Iterative methods....325

Chapter 12 Other capabilities....329
12.1 Automatic time-stepping CFL option....329
12.2 Automatic time-stepping ATS option....330
12.3 Restart analysis....331
12.4 Mapping solution....332
12.5 Skew system....333
12.6 Constraint conditions....334
12.7 Conjugate heat transfer problems....335
12.8 Element birth-death option....336
12.9 Pressure datum....337
12.10 ALE formulation and leader-follower option....339
12.10.1 ALE formulation....339
12.10.2 Leader-follower option....341
12.11 Electro-static and steady current conduction analyses....344
12.11.1 Discretization and method of solution....345
12.11.2 Boundary conditions....345
12.12 Phase change between liquid and vapor....347
12.12.1 General phase change model....347
12.12.2 Cavitation model....349
12.12.3 Phase-change model setup....352
12.13 VOF method....353
12.13.1 Transport equations....353
12.13.2 Material....353
12.13.3 Surface tension....354
12.13.4 VOF-wall angle condition....354
12.13.5 Liquid-solid phase change for primary species....355
12.13.6 Loading of the species....356
12.14 Rigid motion of element group....356

12.15 Steered adaptive meshing 357
12.15.1 Introduction 357
12.15.2 General procedure 360
12.15.3 Criteria for selection of preferable element sizes 365
12.15.4 Combination of criteria 367
12.15.5 Smoothing technique 367
12.15.6 Control on the maximum number of elements 368
12.15.7 Comments and limitations 368
12.15.8 Adaptive Meshing Modeling Tips 370
12.16 Reynolds Equation for lubrication 371
12.16.1 Reynolds Equations for smooth boundaries 371
12.16.2 Reynolds Equations for rough boundaries 372
12.16.3 Modeling guidelines 375
12.17 Universal Barotropic Cavitation model 377

Chapter 13 Other topics 381
13.1 Model preparation and testing 381
13.1.1 Choosing the fluid model 381
13.1.2 Choosing the analysis type 385
13.1.3 Choosing the time/load step size 388
13.1.4 Choosing the computational domain 390
13.1.5 Choosing proper boundary conditions 393
13.1.6 Choosing the proper element 397
13.1.7 Choosing the material model 398
13.1.8 Choosing the solver 398
13.1.9 Model testing 399
13.2 Moving mesh control 402
13.3 Strategies toward obtaining converged solutions 405
13.3.1 Incremental solution procedure 408
13.3.2 Improving initial conditions using restart runs 411
13.3.3 Improving the initial conditions using the mapping file 413
13.3.4 Improving the matrix conditioning using the CFL option 413
13.3.5 Improving the matrix conditioning using proper units 414
13.4 Consistent units, SI units and USCS units 417
13.5 Use of memory and disk 419
13.5.1 Memory usage 419
13.5.2 Disk usage 424

13.6 Brief summary of recommendations for solutions 425
For low Re incompressible flows 426
For high Re incompressible flows 427
Slightly compressible flows 427
High-speed and low-speed compressible flows 427
Ref. 427

Appendix-1 List of figures 429

Appendix-2 List of tables 434

Appendix-3 List of error/warning messages 435

Index 535

Nomenclature

Typical units used here are:

Time	= s , second
Length	= m , meter
Mass	= kg , kilogram
Temperature	= K , Kelvin
Velocity	= $\mathrm{m/s}$, meters per second
Density	= $\mathrm{kg/m^3}$, kilogram per cubic meter
Pressure	= Pa = $\mathrm{N/m^2}$, Pascal
Force	= N = $\mathrm{kg\text{-}m/s^2}$, Newton
Energy	= J = N-m, Joule
Power	= W = $\mathrm{N\text{-}m/s}$, Watt
Electric potential	= V, Volt
[.]	= dimensionless

Scales and reference datums used here in nondimensional quantities are:

L^*	= length scale
V^*	= velocity scale
D^*	= density scale
C^*	= specific heat scale
T^*	= temperature scale
Φ^*	= mass-ratio scale
$\mathbf{x}_r^*$	= (x_r^*, y_r^*, z_r^*) , reference vector of coordinates
θ_r^*	= reference temperature
φ^*	= electric potential scale ($\equiv 1$ here)

Notation	Explanation	Typical unit	Nondimensional form
b	$\frac{1}{2}\mathbf{v}\bullet\mathbf{v}$, specific kinetic energy	$\left[\mathrm{m}^2/\mathrm{s}^2\right]$	b/V^{*2}
Bo	$D^{*2}\|\mathbf{g}\|L^{*3}\beta T^* C^{*2}/k^2$, Boussinesq number	[.]	Bo
Br	$\mu V^{*2}/kT^*$, Brinkman number	[.]	Br
c	$\sqrt{\gamma p/\rho}$, sound speed	$\left[\mathrm{m/s}\right]$	c/V^*
c_1	empirical constant in K-ε turbulence model	[.]	c_1
c_1, c_2	parameters for temperature units such that $c_1\theta + c_2$ is in an absolute temperature scale		
c_2	empirical constant in K-ε turbulence model	[.]	c_2
c_3	empirical constant in K-ε turbulence model	[.]	c_3
c_μ	empirical constant in K-ε turbulence model	[.]	c_μ
Ca	$\mu V^*/\sigma$, Capillary number	[.]	Ca
C_p	specific heat at constant pressure	$\left[\mathrm{m}^2/\mathrm{s}^2\text{-K}\right]$	C_p/C^*
C_v	specific heat at constant volume	$\left[\mathrm{m}^2/\mathrm{s}^2\text{-K}\right]$	C_v/C^*
C^*	specific heat scale	$\left[\mathrm{m}^2/\mathrm{s}^2\text{-K}\right]$	1
d	diffuse reflectivity of specular radiation	[.]	d
d_w^+	empirical constant in turbulence models	[.]	d_w^+

Notation	Explanation	Typical unit	Nondimensional form
D	$\sqrt{\mathbf{e}\otimes\mathbf{e}} \equiv \sqrt{e_{ij}e_{ij}}$, deformation rate (or shear rate)	$[1/\mathrm{s}]$	DL^*/V^*
$\mathbf{D}$	diffusion coefficient tensor in mass transfer equations	$[\mathrm{kg/m\text{-}s}]$	$\mathbf{D}/(D^*V^*L^*)$
D^*	density scale	$[\mathrm{kg/m^3}]$	1
e	specific internal energy	$[\mathrm{m^2/s^2}]$	e/V^{*2}
e	emittance of specular radiation	[.]	e
$\mathbf{e}$	$\frac{1}{2}\left(\tilde{\mathrm{N}}\mathbf{v}+\tilde{\mathrm{N}}\mathbf{v}^T\right)$, velocity strain tensor	$[1/\mathrm{s}]$	$\mathbf{e}L^*/V^*$
$\mathbf{e}_i$	axis vector of moving Cartesian frame	[.]	$\mathbf{e}_i$
E	$\frac{1}{2}\mathbf{v}\bullet\mathbf{v}+e$, specific total energy	$[\mathrm{m^2/s^2}]$	E/V^{*2}
$\mathbf{E}_i$	axis vector of fixed Cartesian frame	[.]	$\mathbf{E}_i$
E_r	$\frac{1}{2}\mathbf{v}_r\bullet\mathbf{v}_r+e$, variable used as "specific total energy" in axisymmetric planes	$[\mathrm{m^2/s^2}]$	E_r/V^{*2}
Ec	V^{*2}/C^*T^*, Eckert number	[.]	Ec
f	view factor in radiation heat transfer	[.]	f
f	vapor fraction	[.]	f
$\mathbf{f}_c$	centrifugal force or Coriolis force per unit volume	$[\mathrm{N/m^3}]$	$\mathbf{f}_cL^*/(D^*V^{*2})$
$\mathbf{f}^B$	body force per unit volume	$[\mathrm{N/m^3}]$	$\mathbf{f}^BL^*/(D^*V^{*2})$

Notation	Explanation	Typical unit	Nondimensional form
$\mathbf{f}_b^B$	buoyancy force per unit volume	$\left[\mathrm{N/m^3}\right]$	$\mathbf{f}_b^B L^* / (D^* V^{*2})$
$\mathbf{f}_g^B$	gravitational force per unit volume	$\left[\mathrm{N/m^3}\right]$	$\mathbf{f}_g^B L^* / (D^* V^{*2})$
$\mathbf{F}$	fluid force	[N]	$\mathbf{F} / (D^* V^{*2} L^{*2})$
$\mathbf{F}$	inviscid flux term in compact form of Navier-Stokes equations		
$\mathbf{g}$	gravitational acceleration vector	$\left[\mathrm{m/s^2}\right]$	$\mathbf{g} L^* / V^{*2}$
$\mathbf{G}$	viscid flux term in compact form of Navier-Stokes equations		
Gr	$D^{*2} \|\mathbf{g}\| L^{*3} \beta T^* / \mu^2$, Grashof number	[.]	Gr
h	heat convection coefficient	$\left[\mathrm{W/m^2\text{-}K}\right]$	$h / (D^* C^* V^*)$
h_i, H_i	shape functions of element variables	[.]	h_i, H_i
h^f	virtual quantity of the variable f	[.]	h^f
H	$E + p/\rho$, enthalpy	$\left[\mathrm{m^2/s^2}\right]$	H / V^{*2}
k	heat conductivity	$\left[\mathrm{W/m\text{-}K}\right]$	$k / (D^* V^* L^* C^*)$
K	kinetic energy of turbulence	$\left[\mathrm{m^2/s^2}\right]$	K / V^{*2}
L^*	length scale	[m]	1
L	latent heat	$\left[\mathrm{m^2/s^2}\right]$	L / V^{*2}
Le	$\|\mathbf{D}\| C^* / k$, Lewis number	[.]	Le
m_b	average bandwidth of the stiffness matrix	[.]	m_b
M	$\|\mathbf{v}\| / c$, Mach number	[.]	M

Notation	Explanation	Typical unit	Nondimensional form
n_b	number of matrix blocks in out-of-core solution	[.]	n_b
n_{bc}	number of blocks while the specular view factor matrix is computed	[.]	n_{bc}
n_{bi}	number of blocks while the specular view factor matrix is read in	[.]	n_{bi}
n_{br}	number of blocks while the specular radiosity matrix is computed	[.]	n_{br}
n_d	number of solution variables (1 to 7 for $u, v, w, p, \theta, K, \varepsilon/\omega$)	[.]	n_d
N_d	number of degrees of freedoms	[.]	N_d
N_e	number of elements	[.]	N_e
N_n	number of nodes	[.]	N_n
N_t	number of time steps	[.]	N_t
N_t^s	number of time steps for which solutions are saved	[.]	N_t^s
N_{es}	maximum number of specular elements in one group	[.]	N_{es}
N_{ns}	maximum number of specular nodes in one group	[.]	N_{ns}
n	unit boundary normal vector outwards from computational domain	[.]	**n**
n	normalized flow direction	[•]	**n**
p	fluid pressure	[Pa]	$p/(D^*V^{*2})$
p_0	pressure datum	[Pa]	$p_0/(D^*V^{*2})$

Notation	Explanation	Typical unit	Nondimensional form
p_k	$p - p_s$, kinetic pressure	[Pa]	$p_k / (D^* V^{*2})$
p_s	$\rho \mathbf{g} \bullet (\mathbf{x} - \mathbf{x}_0) + p_0$, hydrostatic pressure	[Pa]	$p_s / (D^* V^{*2})$
p_{lv}	evaporation pressure	[Pa]	$p_{lv} / (D^* V^{*2})$
p_{vl}	condensation pressure	[Pa]	$p_{vl} / (D^* V^{*2})$
Pe	$D^* V^* L^* C^* / k$, Peclet number	[.]	Pe
Pr	$\mu C^* / k$, Prandtl number	[.]	Pr
$\mathbf{q}$	$-k \tilde{\nabla} \theta$, heat flux vector	$[\mathrm{W/m^2}]$	$\mathbf{q} / (D^* V^* T^* C^*)$
q^B	rate of heat generated per unit volume	$[\mathrm{W/m^3}]$	$q^B L^* / (D^* V^* T^* C^*)$
Q	heat	$[\mathrm{W}]$	$Q / (D^* V^* T^* C^* L^{*2})$
r	first local coordinate in element	[.]	r
r_f	relative distance from fluid FSI nodes to solid FSI interface	[.]	r_f
r_s	relative distance from solid FSI nodes to fluid FSI interface	[.]	r_s
R	heat resistance	$[\mathrm{m^2\text{-}K/W}]$	$R D^* C^* V^*$
Re	$D^* V^* L^* / \mu$, Reynolds number	[.]	Re
Ra	$D^{*2} \lVert \mathbf{g} \rVert L^{*3} \beta T^* C^* / \mu k$, Rayleigh number	[.]	Ra
s	specular reflectivity of radiation	[.]	s
s	second local coordinate in element	[.]	s

Notation	Explanation	Typical unit	Nondimensional form
S	free surface position normal to original free surface	[m]	S/L^*
Sc	$\mu/\|\mathbf{D}\|$, Schmidt number	[.]	Sc
t	time	[s]	tV^*/L^*
t	third local coordinate in element	[.]	t
t	transmittance of specular radiation	[.]	t
Δt	time step size	[s]	$\Delta t V^*/L^*$
Δt_c	critical time step size	[s]	$\Delta t_c V^*/L^*$
T^*	temperature scale	[K]	$1-\theta_r^*/T^*$
u	x-component velocity, or normal velocity	[m/s]	u/V^*
v	y-component velocity, or tangential velocity	[m/s]	v/V^*
$\mathbf{v}$	velocity vector	[m/s]	$\mathbf{v}/V^*$
v^+	dimensionless tangential velocity in turbulent wall functions	[.]	v^+
v_*	shear velocity in near wall regions	[m/s]	v_*/V^*
V^*	velocity scale	[m/s]	1
w	z-velocity, or second tangential velocity, or wave speed	[m/s]	w/V^*
$\mathbf{w}$	mesh velocity vector	[m/s]	$\mathbf{w}/V^*$
w_n	averaged mesh velocity on the interface of control volumes	[m/s]	w_n/V^*
x	first coordinate of position vector	[m]	$(x-x_r^*)/L^*$

Notation	Explanation	Typical unit	Nondimensional form
$\mathbf{x}$	position vector	[m]	$(\mathbf{x}-\mathbf{x}_r^*)/L^*$
$\mathbf{x}_0$	center of rotational coordinate frame, or reference datum location	[m]	$(\mathbf{x}_0-\mathbf{x}_r^*)/L^*$
x_i	coordinates of position vector	[m]	$(x_i-x_r^*)/L^*$
x_r^*	reference datum of x-coordinate	[m]	0
$\mathbf{x}_r^*$	reference datum of coordinate vector	[m]	$\mathbf{0}$
y	second coordinate of position vector	[m]	$(y-y_r^*)/L^*$
y^+	dimensionless normal distance near walls	[.]	y^+
Δy^+	dimensionless normal size of near-wall element	[.]	Δy^+
y_w^+	dimensionless near-wall location used in turbulent wall boundary conditions	[.]	y_w^+
y_r^*	reference datum of y-coordinate	[m]	0
z	third coordinate of position vector	[m]	$(z-z_r^*)/L^*$
z_r^*	reference datum of z-coordinate	[m]	0
α	$\gamma-1$, γ = ratio of specific heats	[.]	α
α	parameter in time integration α-method	[.]	α
α	empirical constant in K-ω turbulence model	[.]	α

Notation	Explanation	Typical unit	Nondimensional form
α_m	number of neighboring nodes, typically 8, 25, 30 and 125 for elements of node=3, 9, 4 and 27, respectively	[.]	α_m
α_ω	empirical constant in K-ω turbulence models	[.]	α_ω
β	$-(d\rho/d\theta)/\rho$, thermal expansion coefficient in Boussinesq approximation	$[1/\mathrm{K}]$	βT^*
β_i	$(d\rho/d\phi_i)/\rho$, mass expansion coefficient in buoyancy force		$\beta_i \Phi^*$
β_K	empirical constant in K-ω turbulence models	[.]	β_K
β_θ	empirical constant in K-ω turbulence models	[.]	β_θ
β_ω	empirical constant in K-ω turbulence models	[.]	β_ω
δ_{ij}	Kronecker delta	[.]	δ_{ij}
ε	dissipation rate of turbulence	$[\mathrm{m}^2/\mathrm{s}^3]$	$\varepsilon L^*/V^{*3}$
ϕ	fluid potential	$[\mathrm{m}^2/\mathrm{s}]$	$\phi/(V^* L^*)$
ϕ_i	mass-ratio of the *i*-th species	[.]	ϕ_i/Φ^*
φ	Euler angle in cylindrical coordinate system	[.]	φ
φ	porosity of porous media	[.]	φ
φ	electric potential	[V]	φ/φ^*
φ^*	electric potential scale	[V]	1
Φ^*	mass-ratio scale		1

Notation	Explanation	Typical unit	Nondimensional form
γ	C_p/C_v, ratio of specific heats	[.]	γ
μ	fluid viscosity	$[\text{kg/m-s}]$	$\mu/(D^*V^*L^*)$
μ_2^*	λ/μ, ratio of the fluid viscosities	[.]	μ_2^*
κ	$\rho dp/d\rho$, fluid bulk modulus of elasticity	[Pa]	$\kappa/(D^*V^{*2})$
κ_0	von Karman constant in turbulence modelings	[.]	κ_0
$\boldsymbol{\kappa}$	permeability tensor of porous media	$[\text{m}^2]$	$\boldsymbol{\kappa}/L^{*2}$
λ	second fluid viscosity	$[\text{kg/m-s}]$	$\lambda/(D^*V^*L^*)$
λ_τ	stress relaxation factor in fluid-structure interaction problems	[.]	λ_τ
λ_d	displacement relaxation factor in fluid-structure interaction problems	[.]	λ_d
λ_{CFL}	CFL number	[.]	λ_{CFL}
θ	temperature	[K]	$(\theta-\theta_r^*)/T^*$
θ_0	reference temperature in buoyancy force	[K]	$(\theta_0-\theta_r^*)/T^*$
θ_e	environmental temperature	[K]	$(\theta_e-\theta_r^*)/T^*$
θ_{lv}	evaporation temperature	[K]	$(\theta_{lv}-\theta_r^*)/T^*$
θ_{vl}	condensation temperature	[K]	$(\theta_{vl}-\theta_r^*)/T^*$
θ_r^*	reference datum of temperature	[K]	0
ρ	fluid density	$[\text{kg/m}^3]$	ρ/D^*
ρ_s	solid density	$[\text{kg/m}^3]$	ρ_s/D^*
σ	coefficient of surface tension	$[\text{Pa-m}]$	$\sigma/(D^*V^{*2}L^*)$

Notation	Explanation	Typical unit	Nondimensional form
σ	$5.6696\times10^{-8}\,\mathrm{W/m^2\text{-}K^4}$, Stefan-Boltzmann constant	$\left[\mathrm{W/m^2\text{-}K^4}\right]$	$\sigma T^{*3}/(D^*V^*C^*)$
σ	electrical conductivity	$\left[\mathrm{W/m\text{-}V^2}\right]$	$\sigma\varphi^{*2}/(D^*V^{*3}L^*)$
σ_K	empirical constant in turbulence models	[.]	σ_K
σ_ε	empirical constant in K-ε turbulence model	[.]	σ_ε
σ_θ	empirical constant in turbulence models	[.]	σ_θ
σ_ω	empirical constant in K-ω turbulence model	[.]	σ_ω
$\boldsymbol{\tau}$	$-(p+\lambda\nabla\bullet\mathbf{v})\mathbf{I}+2\mu\mathbf{e}$, fluid stress	[Pa]	$\boldsymbol{\tau}/(D^*V^{*2})$
$\boldsymbol{\tau}_k$	$\boldsymbol{\tau}-p_s\mathbf{I}$, kinetic stress, p_s = hydrostatic pressure	[Pa]	$\boldsymbol{\tau}_k/(D^*V^{*2})$
$\boldsymbol{\tau}_m$	$-p\mathbf{I}+\mu\nabla\mathbf{v}$, mathematical formulation of "fluid stress"	[Pa]	$\boldsymbol{\tau}_m/(D^*V^{*2})$
ω	specific dissipation rate of turbulence	$[1/\mathrm{s}]$	$\omega L^*/V^*$
Ω	magnitude of angular velocity vector	$[1/\mathrm{s}]$	$\Omega L^*/V^*$
$\boldsymbol{\Omega}$	angular velocity vector	$[1/\mathrm{s}]$	$\boldsymbol{\Omega}L^*/V^*$

Other notations

Notation	Explanation
i, j, k	used as subscripts indicating directions of Cartesian coordinates; repeated indices also mean summations unless noted otherwise.
f	used as a subscript to indicate a fluid variable when it becomes necessary
s	used as a subscript to indicate a solid variable when it becomes necessary
r	used as a subscript to indicate a relative variable
a	used as a subscript to indicate a variable in axisymmetric cases
$\tilde{a}$	nondimensional representation of a variable a
$\overline{a}$	indicates that the value of a is specified as an input parameter in boundary conditions
$\underline{a}$	indicates that the value of a is defined along fluid-structure interfaces only
t	used as a left superscript to indicate a solution time
$\|\mathbf{X}\|$	unless specified otherwise, it is defined as $\sqrt{\sum_i x_i^2}$, Euclidean vector norm, where x_i is the element of $\mathbf{X}$.

This page intentionally left blank

Chapter 1 Introduction

In this Theory and Modeling Guide, the theoretical bases and guidelines for the use of the ADINA CFD and FSI capabilities are presented. These capabilities are implemented in ADINA-F program code. Hence, we use the term ADINA-F throughout this Guide to refer to CFD and FSI capabilities.

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-F.
- To provide specific references that describe the features of ADINA-F in more detail.

Hence, this manual has been compiled to provide a bridge between the actual practical use of the ADINA-F program 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*, Cambridge, MA: Klaus-Jürgen Bathe, 2006

ADINA-F is a general finite-element/finite-volume code that can be used for analyzing fluid-flow related problems. They can be incompressible, slightly compressible, compressible flows and flows in porous media. Most importantly, solid models created in ADINA can be coupled with any fluid model for analyzing fluid-structure interaction problems.

This manual is organized as follows.

We briefly describe the capabilities of ADINA-F and the structure of this manual in the current chapter.

In Chapter 2, we present theoretical bases of various forms of fluid related equations and their variations that have been implemented in ADINA-F.

In Chapters 3 through 7, we present separately our formulations of the fluid models, including the governing equations, numerical methods, available elements or control volumes, boundary conditions, initial conditions and material models. As an exception, we present the incompressible, slightly compressible and low-speed compressible fluid models altogether in Chapter 3, since many similarities exist in these models.

The formulation of mass transfer is presented in Chapter 8, again including the governing equations, numerical method, available elements, boundary conditions, initial conditions and material models. The mass transfer can be coupled with any fluid model described in Chapters 3-7.

The fundamental theory in fluid-structure interactions is presented in Chapter 9. The physical coupling of the fluid and structure models is described. The iterative (partitioned) and direct (simultaneous) numerical procedures are presented.

In Chapter 10, all elements that have been implemented in ADINA-F are detailed for further reference.

The methods for solving nonlinear equations as well as the associated linear equation solvers are described in Chapter 11.

The rest of the capabilities or options are described in Chapter 12, including the CFL option, ATS option, skew systems, constraint equations, conjugate heat transfer option, element birth-death option, pressure datum, ALE formulation, etc.

In Chapter 13, some guidelines and experiences for using ADINA-F are described. They include the most frequently asked questions, the possible modeling errors that may often occur, strategies on choosing and testing models and on obtaining converged solutions, etc.

In the appendices, the figures and tables are listed. All possible error and/or warning messages that ADINA-F may print are also listed in the order of the message code numbers together with explanations.

Finally, the index can be useful in finding topics quickly.

The following table presents a quick overview of the capabilities that are available in ADINA-F. Some combinations may not be available in certain cases. For more details, check the related chapters.

Table 1-1 Overview of capabilities of ADINA CFD & FSI

Category	Capability
Fluid models Note: Heat transfer must be included in compressible flow models.	incompressible, slightly compressible and low-speed compressible flow models
	high-speed compressible flow model
	K-ε and K-ω turbulence models
	flow through porous media
	solid element groups
	heat transfer
	mass transfer
	electro-static and steady current conduction analyses
	volume of fraction
	liquid-vapor phase change
Coupled models	ADINA solid model: mechanical interaction, porous media interaction, and thermal interaction
	porous medium element groups can be coupled with incompressible, slightly compressible and low-speed compressible flow models
	heat transfer can be included in fluid models
	mass transfer can be included in fluid models
	fluid and heat transfer are coupled with electro-static and steady current conduction
Computational domains	2D planar, 2D axisymmetric and 3D
Analyses	steady-state and transient
Elements or control volumes	Galerkin 2D 3/6/9-node elements
	Galerkin 3D 4/29-node elements
	FCBI and FCBI-C 2D 3/4-node elements
	FCBI and FCBI-C 3D 4/5/6/8-node elements
Material models	Constant, time-dependent, temperature-dependent, pressure-dependent, pressure-temperature-dependent, non-Newtonian, velocity-dependent, user-supplied, etc.
Boundary conditions	prescribed solution variables, zero solution variables, zero flux of solution variables, prescribed rotational velocity, concentrated force load, distributed normal-traction load, field centrifugal load, concentrated heat flow load, distributed heat flux load, heat and mass convections, radiation.
	wall, free surface, fluid-fluid interface, phase-change, fluid-structure interface, gap, specular radiation, thermal resistance, uniform flow, angular velocity, fan, vent, friction, , user-supplied, etc.
	external flow, supersonic at inlet, subsonic at inlet, supersonic at outlet, subsonic at outlet, symmetric.
Solvers for linearized equations	Gauss elimination method (COLSOL)
	sparse solver
	iterative methods (RPBCG, RPGMRES, and AMG solvers)

Category	Capability
Other capabilities	automatically nondimensional procedure
	automatic time-stepping CFL option
	automatic time-stepping ATS option
	skew system
	mapping solution
	restart analysis
	constraint condition
	conjugate heat transfer
	element birth-death option
	pressure datum
	include/exclude hydrostatic pressure
	physical or mathematical formulation
	with or without dissipation

Chapter 2 Various forms of fluid equations

In this chapter, we summarize the governing equations used in various fluid flow models in ADINA-F.

2.1 General conservative Navier-Stokes equations

The motion of a continuous fluid medium is governed by the principles of classical mechanics and thermodynamics. In a fixed Cartesian coordinate frame of reference, they can be expressed in conservative forms for mass, momentums and energy, respectively

$$\begin{aligned}
\frac{\partial \rho}{\partial t} + \nabla\bullet\left(\rho \mathbf{v}\right) &= 0 \\
\frac{\partial \rho \mathbf{v}}{\partial t} + \nabla\bullet\left(\rho \mathbf{v}\mathbf{v} - \boldsymbol{\tau}\right) &= \mathbf{f}^B \\
\frac{\partial \rho E}{\partial t} + \nabla\bullet\left(\rho \mathbf{v} E - \boldsymbol{\tau}\bullet\mathbf{v} + \mathbf{q}\right) &= \mathbf{f}^B\bullet\mathbf{v} + q^B
\end{aligned} \qquad (0.1)$$

where, t is the time, ρ is the density, $\mathbf{v}$ is the velocity vector, $\mathbf{f}^B$ is the body force vector of the fluid medium, $\boldsymbol{\tau}$ is the stress tensor, E is the specific total energy, $\boldsymbol{q}$ is the heat flux and q^B is the specific rate of heat generation. The specific total energy and the stress are defined as

$$\begin{aligned}
E &= \frac{1}{2}\mathbf{v}\bullet\mathbf{v} + e \equiv b + e \\
\boldsymbol{\tau} &= (-p + \lambda\nabla\bullet\mathbf{v})\mathbf{I} + 2\mu\mathbf{e}
\end{aligned}$$

where e is the specific internal energy, b is the specific kinetic energy, p is the pressure, μ and λ are the two coefficients of fluid viscosity and $\mathbf{e}$ is the velocity strain tensor

$$\mathbf{e} = \tfrac{1}{2}\left(\nabla\mathbf{v} + \nabla\mathbf{v}^{\mathrm{T}}\right)$$

The heat flux is assumed to obey Fourier's law of heat conduction

$$\mathbf{q} = -k\nabla\theta$$

where θ is the temperature and k is the heat conductivity coefficient.

One of the most important body forces included in $\mathbf{f}^B$ is the gravitational force

$$\mathbf{f}_g^B = \rho\mathbf{g}$$

where $\mathbf{g}$ is the gravitational acceleration vector.

The above equations are based on the Eulerian approach for the description of the continuum motion: the characteristic properties of the medium are considered as functions of time and space in the frame of reference.

The basic or the independent solution variables of the governing equations are either the primitive variables $(p, \mathbf{v}, \theta)$ or the conservative variables $(\rho, \rho\mathbf{v}, \rho E)$. In ADINA-F, the conservative variables are used as solution variables in high-speed compressible flow models and the primitive variables are used in other flow models.

In order to obtain a closed system for solution variables, additional equations called state equations must be provided to correlate the variables, p, ρ, θ, and e. They are usually provided in the form

$$\rho = \rho(p,\theta), \quad e = e(p,\theta)$$

which can be obtained either from fundamental thermodynamics or directly from the fluid properties.

Based on the types of state equations, various forms of equations can be derived for describing incompressible, slightly compressible and compressible flows.

When different coordinate frames of reference are applied, the governing equations are different too. These frames can be fixed, rotational or in arbitrary Lagrangian-Eulerian coordinate systems.

The governing systems that are applied to geometries of two-dimensional planes, two-dimensional axisymmetric planes and three-dimensional spaces have various representations as well.

When the turbulence of flows is required, additional turbulence equations must be solved. These equations can have various forms too, depending on the flow types, coordinate frames, geometries of computational domains, etc.

The equations for the transportation of neutral species must be presented as well to be associated with the fluid equations.

All these equations are related to each other and discussed in the following sections.

2.2 Nonconservative Navier-Stokes equations

The nonconservative forms for mass, momentums and energy, respectively, can be expressed in a fixed Cartesian coordinate frame of reference as

$$\frac{\partial \rho}{\partial t} + \mathbf{v} \bullet \nabla \rho + \rho \nabla \bullet \mathbf{v} = 0$$

$$\rho \frac{\partial \mathbf{v}}{\partial t} + \rho \mathbf{v} \bullet \nabla \mathbf{v} - \nabla \bullet \boldsymbol{\tau} = \mathbf{f}^B$$

$$\rho C_v \frac{\partial \theta}{\partial t} + \rho C_v \mathbf{v} \bullet \nabla \theta + \nabla \bullet \mathbf{q} = 2\mu D^2 + S_c + q^B \qquad (0.2)$$

where D is the deformation rate (shear rate) and S_c is the heat source due to the fluid compressibility:

$$D = \sqrt{\mathbf{e} \otimes \mathbf{e}} \equiv \sqrt{e_{ij} e_{ij}}$$
$$S_c = \nabla \bullet \mathbf{v}\left(-p + \lambda \nabla \bullet \mathbf{v}\right)$$

The nonconservative form of the momentum equation is obtained by subtracting $\mathbf{v}\bullet$ (mass equation) from the conservative momentum equation, and the nonconservative form of the energy equation is obtained by subtracting $(e-b)\cdot$ (mass equation) and $\mathbf{v}\bullet$ (momentum equation) from the conservative energy equation. In this procedure, the variation of internal energy from thermo-dynamics $de = C_v d\theta$ is used, where C_v is the specific heat under the constant volume condition. In incompressible flows, C_v is equal to the specific heat C_p that is obtained at constant pressure.

2.3 Rotating frame of reference

Consider the rotating frame defined in the figure below, where $\left(\mathbf{E}_1, \mathbf{E}_2, \mathbf{E}_3\right)$ are the base vectors of the fixed Cartesian coordinate system and $\left(X_1, X_2, X_3\right)$ are the coordinates used therein, $\left(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\right)$ are the base vectors of the Cartesian coordinate system of the rotating frame, $\left(x_1, x_2, x_3\right)$ are the coordinates used therein, $\mathbf{x}_0 = \mathbf{x}_0\left(t\right)$ is the origin of the rotating frame, and $\mathbf{\Omega} = \mathbf{\Omega}(t)$ is the angular velocity vector that, without losing generality, is assumed to be in the fixed direction $\mathbf{e}_3$. In order to have a clearer presentation, we further use τ to represent the time in the rotating system. The task here is to transform the Navier-Stokes equations from the fixed Cartesian coordinate system to the rotating Cartesian coordinate system.

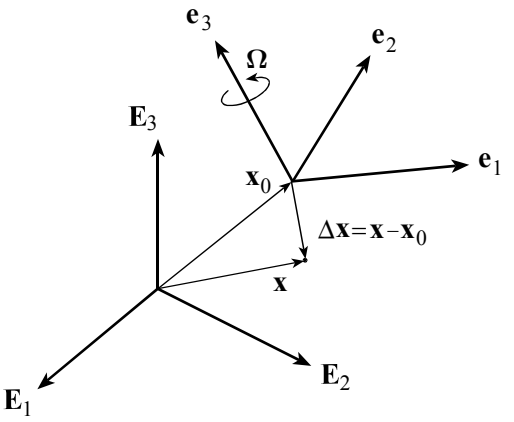


Figure 2.1 Rotating coordinate frame

First, we obtain from the definitions,

$$\mathbf{e}_1'(t) = \Omega(t)\mathbf{e}_2(t)$$
$$\mathbf{e}_2'(t) = -\Omega(t)\mathbf{e}_1(t)$$
$$\mathbf{e}_3'(t) = \mathbf{0}$$

Second, consider an arbitrary fluid particle trajectory expressed in the rotating coordinate system

$$\mathbf{x} = \mathbf{x}_0 + \Delta\mathbf{x} = X_{0i}\mathbf{E}_i + \Delta x_i \mathbf{e}_i$$

Its velocity is, by taking the time derivative,

$$\begin{aligned}\mathbf{v} = \dot{\mathbf{x}} &= \dot{X}_{0i}\mathbf{E}_i + \Delta\dot{x}_i\mathbf{e}_i + \Delta x_i\dot{\mathbf{e}}_i \\ &= \mathbf{v}_0 + \mathbf{v}_r + \boldsymbol{\Omega}\times\Delta\mathbf{x}\end{aligned} \tag{0.3}$$

where $\mathbf{v}_0$ is the velocity of the frame center, $\mathbf{v}_r$ is the velocity with respect to the rotating frame and $\boldsymbol{\Omega}\times\Delta\mathbf{x}$ is the Coriolis velocity resulting from the rotation of the frame. It can be further shown that

$$\partial/\partial t = \partial/\partial\tau - (\mathbf{v} - \mathbf{v}_r)\bullet\nabla$$
$$\nabla\bullet\mathbf{v} = \nabla\bullet\mathbf{v}_r$$
$$\mathbf{v}_r\bullet\nabla\mathbf{v} = \mathbf{v}_r\bullet\nabla\mathbf{v}_r + \boldsymbol{\Omega}\times\mathbf{v}_r$$
$$\mathbf{e} = \tfrac{1}{2}\left(\nabla\mathbf{v}_r + \nabla\mathbf{v}_r^{\mathrm{T}}\right)$$

Using these observations, the nonconservative governing equations in the rotating coordinate system can be expressed as

$$\frac{\partial\rho}{\partial\tau} + \mathbf{v}_r\bullet\nabla\rho + \rho\nabla\bullet\mathbf{v}_r = 0$$
$$\rho\frac{\partial\mathbf{v}_r}{\partial\tau} + \rho\mathbf{v}_r\bullet\nabla\mathbf{v}_r - \nabla\bullet\boldsymbol{\tau} = \mathbf{f}^B + \mathbf{f}_c$$
$$\rho C_v\frac{\partial\theta}{\partial\tau} + \rho C_v\mathbf{v}_r\bullet\nabla\theta + \nabla\bullet\mathbf{q} = 2\mu D^2 + S_c + q^B$$

We can see that these equations are the same as the nonconservative forms of the Navier-Stokes equations by simply replacing the velocity **v** by the relative velocity $\mathbf{v}_r$ and adding the additional force term

$$\mathbf{f}_c = -\rho\ddot{\mathbf{X}}_0 - \rho\dot{\boldsymbol{\Omega}}\times\Delta\mathbf{x} - 2\rho\boldsymbol{\Omega}\times\mathbf{v}_r - \rho\boldsymbol{\Omega}\times\boldsymbol{\Omega}\times\Delta\mathbf{x}$$

It is important to observe that these equations are independent of the coordinate system. We conclude that solving a problem for $(p, \mathbf{v}, \theta)$ is equivalent to solving a problem for $(p, \mathbf{v}_r, \theta)$ with a centrifugal force. The advantage of solving for relative velocity is that we may only deal with a small relative velocity, particularly when the rotational velocity is very large.

It must be remembered, however, that the relative velocity is with respect to the rotating frame. The conditions imposed on relative velocity may be different from those on the velocity. Whenever the velocity is required, it must be calculated according to Eq.(0.3), using the computed relative velocity.

It is further noted that the stress tensor is the same as calculated either using the velocity or the relative velocity. This observation is in fact the theoretical basis that these analysis procedures can directly be applied to fluid domains in fluid-structure interaction analyses.

The equivalent conservative forms of the equations in the rotating frame are

$$\frac{\partial \rho}{\partial \tau} + \nabla \bullet (\rho \mathbf{v}_r) = 0$$

$$\frac{\partial \rho \mathbf{v}_r}{\partial \tau} + \nabla \bullet (\rho \mathbf{v}_r \mathbf{v}_r - \boldsymbol{\tau}) = \mathbf{f}^B + \mathbf{f}_c$$

$$\frac{\partial \rho E_r}{\partial \tau} + \nabla \bullet (\rho \mathbf{v}_r E_r - \boldsymbol{\tau} \bullet \mathbf{v}_r + \mathbf{q}) = (\mathbf{f}^B + \mathbf{f}_c) \bullet \mathbf{v}_r + \mathbf{q}^B$$

where, $E_r = \frac{1}{2}\mathbf{v}_r \bullet \mathbf{v}_r + e$, is the representation of the specific total energy with respect to the rotating frame. Notice that the specific total energy is different from E_r

$$E = E_r + \tfrac{1}{2}(\mathbf{v}_0 + \boldsymbol{\Omega} \times \Delta \mathbf{x}) \bullet (\mathbf{v}_0 + \boldsymbol{\Omega} \times \Delta \mathbf{x} + 2\mathbf{v}_r)$$

This equation indicates that the conditions imposed on the total energy must be modified if the frame is rotating.

2.4 Planar flow

A planar flow is also referred to as a two-dimensional plane flow. In this type of flow, the solution variables do not change on planes parallel to the flow direction. The governing equations for planar flows can be directly deduced from the general forms of the Navier-Stokes equations by choosing a plane that contains the flow trajectories as a coordinate surface. In ADINA-F, the surface is always defined as x_1=const., so that $\partial / \partial x_1 = v_1 = 0$. The resultant equations, in conservative or nonconservative forms, in stationary or rotating frames, are the same as the

equations described in the earlier sections. Of course, the operator ∇ in planar flows must be interpreted as $\mathbf{E}_2 \partial/\partial x_2 + \mathbf{E}_3 \partial/\partial x_3$.

2.5 Axisymmetric flow

An axisymmetric flow is also referred to as a two-dimensional axisymmetric flow. In this type of flow, the solution variables are the same on each plane defined by an arbitrary angle φ in a cylindrical coordinate system (see the figure below). Solutions can therefore be defined in a domain on the (y, z), with y corresponding to the r-coordinate in the cylindrical coordinate system.

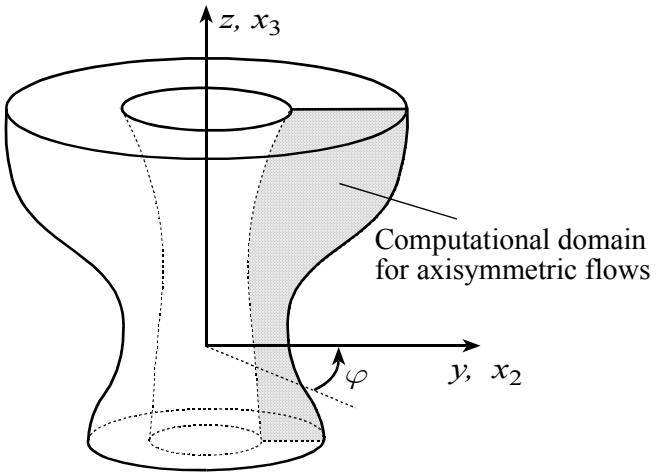


Figure 2.2 Coordinate system for axisymmetric flows

In order to obtain the governing equations for axisymmetric flows, the Navier-Stokes equations are rewritten in the cylindrical coordinate system $(y, \varphi, z) = (x_2, \varphi, x_3)$ and then simplified by using the assumptions φ = const., $\partial/\partial\varphi = 0$ and a zero angular velocity in φ-direction. As in the

planar flow case, we use the notational definition
$\nabla \equiv \mathbf{E}_2 \partial/\partial x_2 + \mathbf{E}_3 \partial/\partial x_3$.

First, the shear rate in the axisymmetric plane can be expressed as

$$\mathbf{e} = \mathbf{e}_a + \Delta\mathbf{e}_a$$

where

$$\mathbf{e}_a = \tfrac{1}{2}\left(\nabla\mathbf{v} + \nabla\mathbf{v}^{\mathrm{T}}\right)$$
$$\Delta\mathbf{e}_a = \mathbf{e}_\varphi \mathbf{e}_\varphi v_2 y^{-1}$$

and the deformation rate is then

$$D = \sqrt{\mathbf{e} \otimes \mathbf{e}} = \sqrt{\mathbf{e}_a \otimes \mathbf{e}_a + \left(v_2 y^{-1}\right)^2}$$

Similarly, the stress tensor can be expressed as

$$\boldsymbol{\tau} = \boldsymbol{\tau}_a + \Delta\boldsymbol{\tau}_a$$

with

$$\boldsymbol{\tau}_a = \left[-p + \lambda y^{-1}\nabla\bullet(y\mathbf{v})\right]\mathbf{I} + 2\mu\mathbf{e}_a$$
$$\Delta\boldsymbol{\tau}_a = 2\mu\Delta\mathbf{e}_a$$

Since the normal boundary direction of any domain on the (y,z) plane is perpendicular to the angular direction $\mathbf{e}_\varphi$, it follows

$$\boldsymbol{\tau}_a\bullet\mathbf{n} = \boldsymbol{\tau}\bullet\mathbf{n}$$

This equation indicates that the stress computed on the (y,z) plane can be applied, as in three-dimensional space, to structural boundaries in fluid-structure interaction analyses.

Finally, we present the axisymmetric flow equations, in a conservative form,

$$\frac{\partial(y\rho)}{\partial t}+\nabla\bullet(y\rho\mathbf{v})=0$$

$$\frac{\partial}{\partial t}(y\rho\mathbf{v})+\nabla\bullet\left[y\left(\rho\mathbf{v}\mathbf{v}-\boldsymbol{\tau}_a\right)\right]=y\mathbf{f}^B+\mathbf{f}_a$$

$$\frac{\partial}{\partial t}(y\rho E)+\nabla\bullet\left[y\left(\rho\mathbf{v}E-\boldsymbol{\tau}_a\bullet\mathbf{v}+\mathbf{q}\right)\right]=y\left(\mathbf{f}^B\bullet\mathbf{v}+q^B\right)$$

and, in a nonconservative form,

$$y\frac{\partial\rho}{\partial t}+\rho\nabla\bullet(y\mathbf{v})+y\mathbf{v}\bullet\nabla\rho=0$$

$$y\rho\frac{\partial\mathbf{v}}{\partial t}+y\rho\mathbf{v}\bullet\nabla\mathbf{v}-\nabla\bullet(y\boldsymbol{\tau}_a)=y\mathbf{f}^B+\mathbf{f}_a$$

$$y\rho C_v\frac{\partial\theta}{\partial t}+y\rho C_v\mathbf{v}\bullet\nabla\theta+\nabla\bullet(y\mathbf{q})=y\left(2\mu D^2+q^B\right)+S_{ca}$$

where

$$S_{ca}=\nabla\bullet(y\mathbf{v})\left[-p+\lambda y^{-1}\nabla\bullet(y\mathbf{v})\right]$$

$$\mathbf{f}_a=\frac{\partial}{\partial\varphi}\left(\boldsymbol{\tau}\bullet\mathbf{e}_\varphi\right)=-\mathbf{e}_2\left[-p+\lambda y^{-1}\nabla\bullet(y\mathbf{v})+2\mu v_2 y^{-1}\right]$$

2.6 Incompressible flow

The compressibility of a fluid, measured as the change of fluid volume or, equivalently speaking, as the change of density under a variation of external pressure, can be represented by the fluid bulk modulus of elasticity $\kappa\left(=\rho\,\partial p/\partial\rho\right)$. The parameter $w\left(=\sqrt{\kappa/\rho}\right)$ represents the wave speed of a small perturbation traveling in the fluid medium. When Bernoulli's

equation $p + \frac{1}{2}\rho \mathbf{v} \bullet \mathbf{v} = \text{const.}$ is applied, the density variation can be approximated by

$$\frac{\Delta\rho}{\rho} = \frac{\Delta p}{\kappa} \approx \frac{|\mathbf{v}|^2}{2w^2}$$

It is therefore clear that a flow can be recognized as incompressible if the fluid speed is much slower than the wave speed. Strictly speaking, the process conditions for which the bulk modulus is defined — whether isothermal, isentropic, etc. — must be designated.

For liquids, these distinctions are not particularly important. Taking water as an example, $\kappa \approx 2 \times 10^9 \,\text{Pa}$, and $\rho \approx 10^3 \,\text{kg}/\text{m}^3$ under standard conditions. This means that the wave speed in water is of the order 1400m/s or 5000km/h. In most water flows, fluid velocities are far less than the wave speed.

For air under isentropic conditions, $p = \text{const.} \cdot \rho^\gamma$ and $w = \sqrt{\gamma p/\rho} = c$, where c is the sound speed and $\gamma \left(\equiv C_p / C_v \approx 1.4 \right)$ is the ratio of specific heats. Therefore, $\Delta\rho/\rho$ is approximately equal to M^2, where M is called the Mach number. For most applications of air flows, $\text{M} < 0.3$ can be used as a critical value to define incompressible flow conditions.

In some applications, the fluid is confined and subjected to an incoming flow rate or deformations of boundaries. The flow must then be treated as "compressible" even though the deformations are small. In these flows, the compressibility of fluids is a physical phenomenon of interest and required to be modeled. These flows are discussed in another section as slightly compressible flows.

In general, incompressible flows are characterized by the state equations

$$\rho = \text{const.}, \quad e = C_v \theta$$

Applying these conditions to the nonconservative forms of the Navier-Stokes equations, the governing equations for incompressible flows and heat transfers are obtained

$$\rho\nabla\bullet\mathbf{v}=0$$

$$\rho\frac{\partial\mathbf{v}}{\partial t}+\rho\mathbf{v}\bullet\nabla\mathbf{v}-\nabla\bullet\boldsymbol{\tau}=\mathbf{f}^{B}$$

$$\rho C_{v}\frac{\partial\theta}{\partial t}+\rho C_{v}\mathbf{v}\bullet\nabla\theta+\nabla\bullet\mathbf{q}=2\mu D^{2}+q^{B}$$

where the stress tensor is also simplified to

$$\boldsymbol{\tau}=-p\mathbf{I}+2\mu\mathbf{e}$$

Let us discuss in more detail the following cases.

- **Mathematical formulation**

Assuming a constant viscosity and applying the mass conservation $\nabla\bullet\mathbf{v}=0$ to the stress tensor, we have

$$\nabla\bullet\boldsymbol{\tau}=\nabla\bullet\boldsymbol{\tau}_{m}$$

where $\boldsymbol{\tau}_{m}=-p\mathbf{I}+\mu\nabla\mathbf{v}$. Using $\boldsymbol{\tau}_{m}$ instead of $\boldsymbol{\tau}$ as "stress" here is purely mathematical since $\boldsymbol{\tau}_{\mathrm{m}}$ differs from the physical stress. In particular, the difference of their normal components on a boundary

$$\left(\boldsymbol{\tau}-\boldsymbol{\tau}_{m}\right)\bullet\mathbf{n}\approx\mu\nabla\left(\mathbf{v}\bullet\mathbf{n}\right)$$

may not be small along open boundaries. However, it can be small enough to be negligible near solid walls. Fluid boundary conditions along most open boundaries comprise either prescribed velocities or applied distributed normal-traction loads. Therefore, in general, the mathematical formulation can be used if no distributed normal-traction conditions are applied. For most practical analyses, mathematical formulation is not recommended.

- **Boussinesq approximation**

Consider a flow field with temperature variations, where the gravitational force is important. In the Boussinesq approximation, the density is assumed constant except for the gravitational force, which is expanded to the first order with respect to the temperature variation

$$\rho = \rho_0 + \frac{\partial \rho}{\partial \theta}(\theta - \theta_0)$$

where ρ_0 is the density at a reference temperature θ_0. Introducing the thermal expansion coefficient

$$\beta = -\frac{1}{\rho_0}\frac{\partial \rho}{\partial \theta}$$

the gravitational force becomes

$$\mathbf{f}_g^B = \rho_0 \mathbf{g}\left[1 - \beta(\theta - \theta_0)\right]$$

For many flows, β can be approximated by an average value near the reference temperature θ_0. When the temperature variation is large in the simulation range, β must be treated as a function of temperature.

Because the values of density that appear in other places in the equations are always assumed constant in the Boussinesq approximation, we will not distinguish the notation ρ from ρ_0 in the Boussinesq approximation.

- **Decoupling of fluid flows from the temperature field**

In the case where the body force $\mathbf{f}^B$ and the fluid constitutive relations do not depend on the temperature, or the temperature variation is small enough to be negligible in the flow field, the continuity equation and momentum equations can be solved without simultaneously solving the energy equation. The temperature field, however, must then be solved with

the calculated fluid velocities. In this case, the fluid flow is decoupled from the temperature field.

- **Effect of dissipation**

The importance of the dissipation term, $2\mu D^2$ in the energy conservation equation depends on the fluid properties and the flows. Let us compare the major terms in the energy equation using some characteristic scales, L^*, V^*, T^* and D^* for length, velocity, temperature and density respectively,

$$\frac{\text{dissipation}}{\text{convection}} \sim \frac{\mu V^*}{C_v D^* T^* L^*} = \frac{\text{Ec}}{\text{Re}}$$

$$\frac{\text{dissipation}}{\text{diffusion}} \sim \frac{\mu V^{*2}}{kT^*} = \text{PrEc}$$

The effect of the dissipation term can then be neglected if both terms are small.

For air where $\text{Pr} \approx 0.7$ and $\text{Ec} \approx \text{M}^2$, the dissipation term can be neglected if the Mach number is very small since the Reynolds number is usually not small.

For oil, on the other hand, the Prandtl number could be as high as 10^4, then whether to drop the dissipation term depends on the Eckert number and the Reynolds number.

Finally, whenever the dissipation is small enough to be negligible, the Eckert number is generally small and the temperature is weakly coupled to the flow field.

- **Hydrostatic pressure**

In a stationary fluid domain, the pressure becomes the hydrostatic pressure that is balanced by the gravitational force

$$p = p_s = \rho \mathbf{g} \bullet \left(\mathbf{x} - \mathbf{x}_0 \right) + p_0$$

where p_0 is a pressure datum at $\mathbf{x}_0$. In general, the hydrostatic pressure p_s is always part of the absolute pressure solution p. The hydrostatic pressure is not coupled with the velocity field and thus "hidden" in the solution procedure. In many flows, the hydrostatic pressure is much larger in magnitude than the kinetic pressure

$$p_k = p - p_s$$

In this case, the presence of the hydrostatic pressure may cause instability in numerical computations. It is then recommended to use the kinetic pressure instead of the absolute pressure as the solution variable. This procedure can be enforced by means of simply removing the hydrostatic pressure term from the gravitational force

$$\mathbf{f}_b^B = \mathbf{f}_g^B - \nabla p_s = -\rho \mathbf{g} \beta \left(\theta - \theta_0\right)$$

The force $\mathbf{f}_b^B$ is called the buoyancy force. Thus, the kinetic pressure replaces the pressure and the buoyancy force replaces the gravitational force. Keep in mind that the "stress" used in the new formulation is not the physical stress. It is called the kinetic stress and does not contain the hydrostatic pressure effect

$$\boldsymbol{\tau}_k = \boldsymbol{\tau} + p_s \mathbf{I}$$

Hence, if this formulation is applied to fluid-structure interaction problems, the hydrostatic pressure must be applied separately as an additional force to structures.

2.7 Fluid flow through porous media

A porous medium saturated with a fluid has very different properties than a pure fluid medium (see the following figure for a typical saturated porous medium). The porous domain consists of both fluid and solid. In the case where the solid is rigid, fluid flows depend only on the properties

of the fluid region that is formed by many small holes of possibly different shapes and sizes. Although the Navier-Stokes equations are valid for flows through porous media, the simulation of fluid flows through these regions of small scales is impractical and far beyond the capacity of present day computers if a large domain is considered.

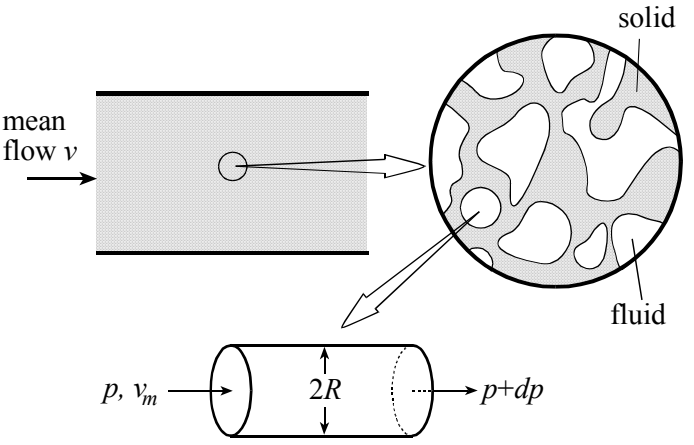


Figure 2.3 Illustration of porous media models

In order to determine the flow properties in porous media, the mean (or averaged) velocity must be used. During the averaging procedure, an important property of the porous medium, called permeability, is introduced.

Let us consider a simple example as shown in the figure above, where viscous fluid flows through a micro-circular hole are subjected to a constant pressure drop. The equation that governs the flow can be obtained from momentum equation

$$\frac{dp}{dx} = \frac{\mu}{y}\frac{\partial}{\partial y}\left(y\frac{\partial v_m}{\partial y} \right)$$

where v_m is the velocity of fluid in the hole. By imposing the no-slip condition on the wall, the exact solution can be obtained

$$v_m = \frac{1}{4\mu}\frac{\partial p}{\partial x}\left(y^2 - R^2\right)$$

Integrating the solution along the cross section of the hole, we obtain the averaged velocity through the hole

$$\hat{v} = \frac{1}{\pi R^2}\int_0^{2\pi}\int_0^R v_m y dy d\varphi = -\frac{R^2}{8\mu}\frac{\partial p}{\partial x}$$

Considering that such holes are sparsely located in a porous medium that has a porosity ϕ (the ratio of the volume occupied by the fluid to the volume of the mixed medium), the averaged velocity in that medium is

$$v = \phi\hat{v} = -\frac{\phi R^2}{8\mu}\frac{\partial p}{\partial x}$$

The coefficient $\phi R^2/8$ here is called the permeability of the porous medium discussed in this example. It is clear then that the permeability has units of square length.

Of course, the actual permeability of a porous medium depends on the shapes of the saturated region, that could differ from each other even within the same porous medium. Furthermore, the permeability may be anisotropic and flow dependent. Nevertheless, a more general form that governs flows through porous media can be expressed as

$$\beta_F \rho \left\|\mathbf{v}\right\| \mathbf{v} + \mu\boldsymbol{\kappa}^{-1}\bullet\mathbf{v} = -\nabla p + \mathbf{f}^B \qquad (0.4)$$

where

$$\boldsymbol{\kappa} = \kappa_{ij}\mathbf{E}_i\mathbf{E}_j$$

is the permeability tensor and β_F is the inertia factor. Eq.(0.4) is called Darcy's law or the Darcy-Forchheimer equation. It governs the momentum conservation of fluid flows through porous media.

The continuity equations obtained in all sections are valid for the averaged velocity, which controls the mass conservation of the fluid in the porous medium considered. In other words, the porous media flows can be assumed to be incompressible, slightly compressible or compressible.

The energy equations obtained in the previous and later sections are all valid as well, in both fluid and solid regions in porous media. In order to obtain an averaged temperature solution, the material properties are averaged using fluid and solid properties

$$\rho C_v = \phi\left(\rho_f C_{vf}\right) + \left(1-\phi\right)\left(\rho_s C_{vs}\right)$$
$$k = \phi k_f + \left(1-\phi\right)k_s$$

The continuity equation, Darcy's law (replacing the momentum equations) and the energy equation (using averaged material data) form the governing equation system for fluid flows through porous media.

The Darcy's law equation is different from the Navier-Stokes equations. For instance, there is no derivative of the velocity. Some conventional no-slip wall boundary conditions may not be appropriate for porous media flows. For example, consider a saturated porous medium in a pipe subjected again to a constant pressure drop as shown in Fig.2.3. The size of the pipe here of course is much larger than that of the micro-holes discussed earlier. There would be no solution if a no-slip condition were applied on the wall. Instead, Darcy's law gives the exact solution if a slip wall condition is applied.

Also note that Darcy's law has no time derivatives. Therefore, if the body force $\mathbf{f}^B$ is constant, it is unnecessary to perform transient analyses. Many fluid flow problems in porous media are in fact coupled with temperature using the Boussinesq approximation to the gravitational force.

Finally, for a better understanding of Darcy's equation, let us apply the mass conservation condition to it,

$$\nabla\bullet\boldsymbol{\kappa}\bullet\nabla p = \nabla\bullet\boldsymbol{\kappa}\bullet\mathbf{f}^B$$

This is a type of Poisson equation for pressure. The pressure is then very much like a velocity potential.

The pressure in porous media sometimes is also called pore pressure. When the deformability of the solid in porous media is considered, the

velocity in Darcy's equation becomes the relative velocity $\mathbf{v} - \mathbf{w}$, where $\mathbf{w}$ is the velocity of the structure.

2.8 Slightly compressible flow

Any fluid exhibits some compressibility. Even some fluids conventionally classified as incompressible, like water and oil, exhibit the compressibility that cannot be ignored, for example, in confined flows. The compressibility of a fluid is characterized by its bulk modulus of elasticity κ. As discussed in Section 2.6, when the fluid velocity is much smaller than the wave velocity, the flow can generally be treated as incompressible.

However, the compressibility can be a key factor in so-called "incompressible" fluids. For example, when a fluid is subjected to a sudden load of disturbance, e.g., a load of pressure, a very high fluid pressure can quickly build up. Whenever the propagation of a pressure wave is to be calculated, or the fluid domain is enclosed by deformable boundaries, the compressibility must be included. This fluid response to such a system is considered in many applications of fluid and fluid-structure interaction problems, such as shock absorbers and air bags in the automobile industry, traveling of pressure waves in explosions, simulation of noise, etc.

A slightly compressible flow model is suitable for such problems. It can be identified by the state equations

$$\rho = \rho_0 \left(1 + \frac{p}{\kappa}\right), \quad e = C_v \theta \tag{0.5}$$

where ρ is the fluid density with the compressibility and ρ_0 is the density at $p = 0$. The conservative form of continuity equation remains the same as that in the general case. The nonconservative form of the continuity equation in slightly compressible flows can be written as

$$\frac{\rho_0}{\kappa}\left(\frac{\partial p}{\partial t} + \mathbf{v} \bullet \nabla p\right) + \rho \nabla \bullet \mathbf{v} = 0$$

In the axisymmetric case, this equation becomes

$$\frac{y\rho_0}{\kappa}\left(\frac{\partial p}{\partial t}+\mathbf{v}\bullet\nabla p\right)+\rho\nabla\bullet\left(y\mathbf{v}\right)=0$$

The momentum and energy equations for incompressible flows are still valid.

To be able to further understand the behavior of the slightly compressible model, let us consider the special case where the velocity is negligibly small, $\mathbf{f}^B=\mathbf{0}$ and κ is a large constant. The continuity and momentum equations can be simplified to

$$\frac{\rho_0}{\kappa}\frac{\partial p}{\partial t}+\rho_0\nabla\bullet\mathbf{v}=0$$
$$\rho_0\frac{\partial \mathbf{v}}{\partial t}+\nabla p=0$$

Fluid flows governed by these equations are called acoustic flows. They can be further simplified by introducing the fluid potential ϕ,

$$\mathbf{v}=\nabla\phi,\ \ p=-\rho_0\frac{\partial\phi}{\partial t}$$

to

$$\frac{\partial^2\phi}{\partial t^2}=w^2\nabla^2\phi \qquad (0.6)$$

where w is the wave speed. Eq.(0.6) is called the acoustic equation. This equation can be solved in general using the ADINA program (see "Theory and Modeling Guide, Volume I: ADINA Solids & Structures").

We can see that the slightly compressible flow model is a more general version of acoustic flows.

2.9 Compressible flows

A typical fluid medium of compressible flows is air. The ideal gas law

$$p = \left(C_\rho - C_v\right)\rho\theta, \quad e = C_v\theta \tag{0.7}$$

with constants specific heats C_p and C_v is usually used in the simulation of gas flows. Under most conditions, the specific heats depend only on temperature although they may depend on pressure as well in rare occasions.

The physical phenomenon of compressible flows differs considerably from incompressible flows. A most important difference is the interdependence of pressure, temperature and density. This dependency results in a strong coupling of the momentum and energy equations, and presents a highly nonlinear behavior in solutions.

The most important parameter that characterizes the difference is the Mach number M. The flow is classified as subsonic when $M < 1$, supersonic when $M > 1$ and transonic if the flow varies between the subsonic and supersonic regimes. As soon as the Mach number exceeds the magnitude one, shock waves, strong rarefactions, sonic surfaces, and contact discontinuities begin to appear. These phenomena present serious difficulties in computational fluid dynamics. Therefore, in general, the methods that are used to analyze incompressible flows are not effective for high-speed compressible flows. Furthermore, some special boundary conditions that deal with these nonlinear behaviors and coupling of variables are required uniquely for each flow regime along the boundaries.

On the other hand, when the Mach number is very small, $M \ll 1$, the flows behave much like incompressible flows and yet the methods that are effective for high-speed compressible flows may not be appropriate or applicable to flows in this regime. The methods that are suitable for incompressible flows may then be applied to this type of flow even if the compressibility is included.

We will, therefore, distinguish compressible flows for high- and low-speed regimes. In our approach, the state equation for compressible flows is given in Eq.(0.7). The conservative form of the continuity equation is used for all compressible flows. For high-speed flows, the conservative equations are used, and the solution variables are conservative variables

$(\rho, \rho\mathbf{v}, \rho E)$. For low-speed flows, conservative equations are used for FCBI and FCBI-C elements while nonconservative equations are used other types of elements. The solution variables are primitive variables $(p, \mathbf{v}, \theta)$.

If FCBI-C element is used, there is an option for using either the total energy equation (0.1) or the heat transfer equation (0.2). In addition to a general temperature dependent ideal gas law, some real gas models are also available for FCBI-C element with the total energy equation: the standard Redlich Kwong model, the Aungier Redlich Kwong model, the Soave Redlich Kwong model and the Peng Robinson model.

2.10 High-speed compressible flow

The conservative forms of compressible flow equations can be written in a compact form

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \bullet \left[\mathbf{F}(\mathbf{U}) + \mathbf{G}(\mathbf{U}) \right] = \mathbf{C}$$

where

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho\mathbf{v} \\ \rho E \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} \mathbf{0} \\ -\boldsymbol{\sigma} \\ -\boldsymbol{\sigma} \bullet \mathbf{v} + \mathbf{q} \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 0 \\ \mathbf{f}^B \\ \mathbf{f}^B \bullet \mathbf{v} + q^B \end{bmatrix} \tag{0.8}$$

$$\mathbf{F}(\mathbf{U}) \equiv \mathbf{F}(\mathbf{v}, \mathbf{U}) = \mathbf{v}\mathbf{U} + \begin{bmatrix} \mathbf{0} \\ p\mathbf{I} \\ \mathbf{v}p \end{bmatrix}$$

and $\boldsymbol{\sigma}$ and H are, respectively, the shear stress and the enthalpy

$$\boldsymbol{\sigma} = \boldsymbol{\tau} + p\mathbf{I} = \lambda \nabla \bullet \mathbf{v}\mathbf{I} + 2\mu\mathbf{e}$$

$$H = E + \frac{p}{\rho}$$

Note that we have expressed $\mathbf{F}$ as a function of $\mathbf{v}$ and $\mathbf{U}$. The argument $\mathbf{v}$ here represents only the convective velocity defined in the first term on the right-hand side of Eq.(0.8). This notation will be more convenient later in the discussion on the ALE formulation.

When viscosity and heat conductivity are neglected ($\mathbf{G} = \mathbf{0}$), the flow is called inviscid and the governing equations are the Euler equations. Consider the flux vector in a direction $\mathbf{n}$, $\mathbf{F}_n = \mathbf{n} \bullet \mathbf{F}$, subject to a variation of the variable $\Delta\mathbf{U}$,

$$\Delta\mathbf{F}_n = \mathbf{A}\Delta\mathbf{U}$$

where $\mathbf{A} = \partial\mathbf{F}_n / \partial\mathbf{U}$ is the Jacobian of the flux function $\mathbf{F}_n$. The matrix $\mathbf{A}$ can be further written in terms of its eigenvectors and eigenvalues

$$\mathbf{A} = \mathbf{P}\mathbf{D}\mathbf{P}^{-1}$$

where the column vectors of the matrix $\mathbf{P}$ are right eigenvectors, the row vectors of the matrix $\mathbf{P}^{-1}$ are left eigenvectors and $\mathbf{D}$ is the diagonal matrix which consists of the eigenvalues

$$\mathbf{P} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ \mathbf{v} + c\mathbf{n} & \mathbf{v} - c\mathbf{n} & \mathbf{v} & \boldsymbol{\tau}_1 & \boldsymbol{\tau}_2 \\ H + uc & H - uc & H - c^2/\alpha & v_1 & v_2 \end{bmatrix} \tag{0.9}$$

$$\mathbf{P}^{-1} = \frac{1}{2c^2} \begin{bmatrix} -uc + \alpha b & c\mathbf{n} - \alpha\mathbf{v} & \alpha \\ uc + \alpha b & -c\mathbf{n} - \alpha\mathbf{v} & \alpha \\ 2c^2 - 2\alpha b & 2\alpha\mathbf{v} & -2\alpha \\ -2c^2 v_1 & 2c^2\boldsymbol{\tau}_1 & 0 \\ -2c^2 v_2 & 2c^2\boldsymbol{\tau}_2 & 0 \end{bmatrix} \tag{0.10}$$

$$\mathbf{D}(u,c) = \text{diag}\{u+c, u-c, u, u, u\} \qquad (0.11)$$

where, $(\mathbf{n}, \boldsymbol{\tau}_1, \boldsymbol{\tau}_2)$ form a locally orthogonal coordinate system and (u, v_1, v_2) are the corresponding velocity components.

When the flow is steady, $\partial/\partial t = 0$, we can see that the Euler equations are singular at locations where $\mathbf{v} = \mathbf{0}$. Such a position is called a stagnation point. Hence, if the Mach number becomes very small, the system becomes sensitive and eventually becomes singular when the Mach number approaches zero.

Furthermore, we can see that waves propagate at different speeds (represented by the eigenvalues defined in Eq.(0.11)) in their corresponding directions (represented by the eigenvectors defined in Eq. (0.9)). Special boundary conditions are therefore required to deal with such flow complexities. These conditions are discussed in Chapter 6.

2.11 Turbulence models

In fluid mechanics, the flows of practical relevance are almost always turbulent; this means that the fluid motion is highly random, unsteady and three-dimensional. Due to these complexities, the turbulent motion and the associated heat and mass transfer phenomena are extremely difficult to describe and to present theoretically and numerically. Theoretically speaking, the Navier-Stokes equations are still valid for describing turbulent flows. However, the storage capacity and speed of present day computers are far from sufficient to allow a solution for any practically relevant turbulent flow. The reason is that the turbulent motion contains scales that are much smaller than the extent of the flow pattern described by the averaged variables. On the other hand, when the speed of fluid flow is not very high, the random motion of fluid does not contribute much energy to the mainstream fluid flow compared with that of averaged variables. Only when such random motions of fluid become too strong to be ignored, they must be modeled into the stream of averaged variables. Generally, a high Reynolds number characterizes turbulent flows.

Many turbulence models have been developed to predict the effect of these random flow motions to the flow patterns of regularly large scales. In these models, the viscosity and heat conductivity coefficient are modified to

$$\mu = \mu_0 + \mu_t, \quad k = k_0 + k_t$$

where μ_0 and k_0 are, respectively, the laminar viscosity and the laminar heat conductivity, while μ_t and k_t are their turbulence counterparts, which are computed as described in the following models.

- **Large-eddy-simulation model for incompressible flows**

The large-eddy-simulation model is an algebraic model, where μ_t and k_t are computed by directly defined functions

$$\mu_t = \sqrt{2}\rho k_D^2 \Lambda^2 D, \quad k_t = \frac{\mu_t C_p}{\mathrm{Pr}}$$

in which D is the effective deformation rate, k_D is a dimensionless model constant, Λ is the element size and Pr is the turbulent Prandtl number.

- **K-ε turbulence model in incompressible flows**

In this model, μ_t and k_t are formed as

$$\mu_t = \rho c_\mu \frac{K^2}{\varepsilon}, \quad k_t = \frac{\mu_t C_p}{\sigma_\theta}$$

where K and ε, being called kinetic energy and rate of dissipation of the turbulence respectively, are defined as

$$K = \frac{1}{2}\overline{\mathbf{v}' \bullet \mathbf{v}'}, \quad \varepsilon = \frac{\mu_0}{\rho}\overline{(\nabla \mathbf{v}') \otimes (\nabla \mathbf{v}')}$$

Here $\mathbf{v}'$ is the fluctuating velocity or the velocity difference between the turbulent velocity and the averaged velocity. K and ε are additional variables governed by two additional equations

$$\frac{\partial\left(y^a \rho K\right)}{\partial t} + \nabla\bullet\left[y^a\left(\rho \mathbf{v} K - \mathbf{q}_K\right)\right] = y^a S_K$$
$$\frac{\partial\left(y^a \rho \varepsilon\right)}{\partial t} + \nabla\bullet\left[y^a\left(\rho \mathbf{v} \varepsilon - \mathbf{q}_\varepsilon\right)\right] = y^a S_\varepsilon$$

where a is zero for two- and three-dimensional flows and one for axisymmetric flows. Other corresponding terms are defined as

$$\mathbf{q}_K = \left(\mu_0 + \frac{\mu_t}{\sigma_K}\right)\nabla K$$
$$\mathbf{q}_\varepsilon = \left(\mu_0 + \frac{\mu_t}{\sigma_\varepsilon}\right)\nabla \varepsilon$$
$$S_K = 2\mu_t D^2 - \rho\varepsilon + B$$
$$S_\varepsilon = \frac{\varepsilon}{K}\left[2c_1\mu_t D^2 - c_2\rho\varepsilon + c_1\left(1-c_3\right)B\right]$$
$$B = \left(\mu_0 + \frac{\mu_t}{\sigma_\theta}\right)\beta\mathbf{g}\bullet\nabla\theta$$

Here, c_μ, c_1, c_2, c_3, σ_K, σ_ε, and σ_θ are model constants.

The K-ε turbulence model turbulence model: K-ε modelis applicable to certain high Reynolds number flows.

- **K-ε RNG turbulence model in incompressible flows**

The K-ε RNG model uses the same formulae as the turbulence K-ε model except the coefficient c_1 in the ε-equation is modified to

$$c_1 - \frac{A\left(1 - A/4.38\right)}{1 + 0.015A^3} \rightarrow c_1$$

where $A = \sqrt{2}KD/\varepsilon$. The K-ε RNG turbulence model is also applicable to certain high Reynolds number flows.

- **K-ω high Reynolds number turbulence model in incompressible flows**

The variable ω, initially introduced purely from a mathematical point of view, is called the specific dissipation rate of turbulence. The variable ω is related to K and ε as

$$\omega \sim \frac{\varepsilon}{K}$$

Then μ_t and k_t are computed by

$$\mu_t = \alpha\rho\frac{K}{\omega}, \quad k_t = \frac{\mu_t C_p}{\sigma_\theta}$$

The governing equations for K and ω are

$$\frac{\partial\left(y^a\rho K\right)}{\partial t} + \nabla\bullet\left[y^a\left(\rho\mathbf{v}K - \mathbf{q}_K\right)\right] = y^a G_K$$
$$\frac{\partial\left(y^a\rho\omega\right)}{\partial t} + \nabla\bullet\left[y^a\left(\rho\mathbf{v}\omega - \mathbf{q}_\omega\right)\right] = y^a G_\omega$$

where a is zero for two- and three-dimensional flows and one is for axisymmetric flows. Other corresponding terms are defined as

$$\mathbf{q}_\omega = \left(\mu_0 + \frac{\mu_t}{\sigma_\omega}\right)\nabla\omega$$
$$G_K = 2\mu_t D^2 - \beta_K\rho K\omega + B$$
$$G_\omega = \frac{\omega}{K}\left(2\alpha_\omega\mu_t D^2 - \beta_\omega\rho K\omega + \beta_\theta B\right)$$

Here, α, α_ω, β_K, β_ω, σ_K, σ_ω, σ_θ and β_θ are model constants.

- **K-ω low Reynolds number turbulence model in incompressible flows**

The basic equations for the low Reynolds number model are the same as those for the high Reynolds number model. However, the following parameters are used

$$\alpha = \alpha^h \frac{1/40 + R_K}{1 + R_K}, \quad \alpha_\omega = \alpha_\omega^h \frac{1/10 + R_\omega}{1 + R_\omega} \alpha^{-1}, \quad \beta_K = \beta_K^h \frac{5/18 + R_\beta}{1 + R_\beta}$$

where the superscript h indicates the values defined in the high Reynolds number model and

$$R_K = \frac{R_t}{6}, \quad R_\omega = \frac{R_t}{2.7}, \quad R_\beta = \left(\frac{R_t}{8} \right)^4, \quad R_t = \rho \frac{K}{\mu_0 \omega}$$

- **K-ε turbulence model in compressible flows**

In this model, the effects of compressibility have been taken into account. The specific total energy is modified to include the kinetic energy of turbulence

$$E = \frac{1}{2} \mathbf{v} \bullet \mathbf{v} + e + K$$

and the stress tensor is also modified to

$$\boldsymbol{\tau} = \left(-p + \lambda \nabla \bullet \mathbf{v} - \frac{2}{3} \rho K \right) \mathbf{I} + 2\mu \mathbf{e}$$

With these modifications, the K-ε equations of the K-ε model in incompressible flows are used.

- **Turbulent wall function**

An important step towards the successful use of the two-equation models, the K-ε model and the K-ω model, is a proper treatment of the fluid-solid boundary conditions. In particular, the dissipation rate ε can become infinite and therefore requires special attention. When boundary conditions are applied directly at the wall, the equations must be integrated through the viscous sub-layer, which is undesirable, since many elements have to be placed in this sub-layer for proper resolution of the high Reynolds number turbulence phenomenon. Instead, extensive research has been conducted and has provided empirical laws of sufficient generality that connect the wall conditions to dependent variables just outside the viscous sub-layer. The universal law of the near-wall velocity profile, also called the turbulent wall function, is the most frequently used law. The so-called Reichardt law is used in ADINA-F

$$v^+ \equiv W(y^+) = \frac{1}{\kappa_0}\ln\left(1+\kappa_0 y^+\right) + 7.8\left(1 - e^{-\frac{y^+}{11}} - \frac{y^+}{11}e^{-\frac{y^+}{3}}\right)$$

where

$$v^+ = \left(v - w\right)/v_*$$
$$y^+ = \rho v_* y/\mu_0$$

v is the velocity component tangential to the wall, w is the tangential velocity of the solid, v_* is the shear velocity, y is the distance from the wall, κ_0 is the von Karman constant, ρ is the fluid density and μ_0 is the laminar viscosity.

- **Shear Stress Transport (SST) model in incompressible flows**

The main idea behind the SST turbulence model is to combine the accuracy of the K-ω model in the near wall region and the free stream independence of the K-ε model. As a result, the SST model delivers an improved separation prediction and no free stream dependency.

The governing equations for the K-ω model apply to the SST model without modifications. However, the eddy viscosity is redefined in the following way:

$$\mu_t = \alpha^s \rho a_1 \frac{K}{\max(a_1\omega, \sqrt{2}DF_2)}$$

Additionally, the source term in the ω equation of the SST model, G_ω^s, has an additional contribution called the cross-diffusion term, C:

$$G_\omega^s = G_\omega + C = G_\omega + 2\rho(1-F_1)\sigma_{\omega_2}(\nabla K)\cdot(\nabla \ln \omega)$$

The functions F_1 and F_2 are designed to be one near the wall and zero for the free shear layers

$$(F_1, F_2) = \left(\tanh\left(\Phi_1^4\right), \tanh\left(\Phi_2^2\right)\right)$$

where

$$\Phi_1 = \min\left\{\max\left\{\frac{\sqrt{K}}{\beta_K \omega y}, \frac{500\mu}{\rho\omega y^2}\right\}, \frac{4\rho\sigma_{\omega_2}K}{C^+ y^2}\right\}$$

$$\Phi_2 = \max\left\{\frac{2\sqrt{K}}{0.09\omega y}, \frac{500\mu}{\rho\omega y^2}\right\}$$

$$C^+ = \max\left\{C, 10^{-10}\right\}$$

The constants used in the SST model $\phi = (\alpha^s, \alpha_\omega^s, \beta_\omega^s, \sigma_K^s, \sigma_\omega^s, ...)$ are calculated using the function F_1 in the following way:

$$\phi = F_1\phi_1 + (1-F_1)\phi_2$$

where ϕ_1 and ϕ_2 are the constants for the K-ω and the K-ε model respectively.

- **Spalart-Allmaras model**

The Spalart-Allmaras (SA) is a one-equation model. The new variable $\tilde{\nu}$, obtained through the SA transport equation is called the modified eddy viscosity that has the same unit as the kinematic viscosity, namely, L^2/t.

The eddy viscosity in the SA model is defined as,

$$\mu_t = \rho\tilde{\nu} f_{v1}$$

where f_{v1} is a non-dimensional function of $\tilde{\nu}$ and will be given later.

The SA transport equation for $\tilde{\nu}$ is,

$$\frac{\partial\left(y^a \rho\tilde{\nu}\right)}{\partial t} + \nabla\bullet\left[y^a\left(\rho\boldsymbol{v}\tilde{\nu} - \boldsymbol{q}_{\tilde{\nu}}\right)\right] = y^a G_{\tilde{\nu}}$$

where

$$\boldsymbol{q}_{\tilde{\nu}} = \frac{1}{\sigma_s}\left(\mu + \rho\tilde{\nu}\right)\nabla\tilde{\nu}$$

$$G_{\tilde{\nu}} = \rho c_{b1}\tilde{S}\tilde{\nu} - \rho c_{w1} f_w\left(\frac{\tilde{\nu}}{d}\right)^2 + \rho\frac{c_{b2}}{\sigma_s}\left(\nabla\tilde{\nu}\right)\bullet\left(\nabla\tilde{\nu}\right)$$

and d is the distance to the closest wall. The related functions are defined as

$$f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3}$$

$$\tilde{S} = S + \frac{\tilde{\nu}}{\kappa^2 d^2}\left(1 - \frac{\chi}{1 + \chi f_{v1}}\right)$$

$$f_w = g\left(\frac{1 + c_{w3}^6}{g^6 + c_{w3}^6}\right)^{1/6}$$

where

$$\chi = \tilde{\nu}/\nu$$

$$S = \omega + C_{prod} \min\left(0, e - \omega\right)$$

$$e = \sqrt{2} D$$

$$\omega = \sqrt{\tfrac{1}{2}\boldsymbol{\omega} \otimes \boldsymbol{\omega}} \qquad \left(\boldsymbol{\omega} = \nabla \mathbf{v} - \left(\nabla \mathbf{v}\right)^T\right)$$

$$g = r + c_{w2}(r^6 - r) \qquad \left(r = \tilde{\nu} \Big/ \left(\tilde{S} \kappa^2 d^2\right)\right)$$

and $c_{w1}, c_{w2}, c_{w3}, \kappa, c_{b1}, c_{b2}, c_{v1}, \sigma_s, C_{prod}$, are model constants.

The wall boundary condition for the SA transport equation is $\tilde{\nu} = 0$. For the momentum equations, the wall function approach or near wall treatment is used to obtain the wall shear stress in the same as in the two-equation models.

The typical inlet or free stream boundary condition is prescribed $\tilde{\nu}$. The desired values should be about $\left(0.033 i \,\mathrm{Re}\right)\mu/\rho$, where μ is the laminar viscosity of the fluid, Re is the inlet Reynolds number, and i $\left(= 0.01 \sim 0.1\right)$ is turbulence intensity. For most free stream conditions, $\tilde{\nu} = 0.1\mu/\rho$ can be used.

- **Detached Eddy Simulation (DES) model**

DES is initially developed to solve problems with large regions of separation and high Reynolds numbers. For such type of flows LES is very expensive and Reynolds-Averaged Navier-Stokes models (RANS) are not accurate enough.

DES uses a one-equation RANS model (in our case the Spalart-Allmaras model) which functions as a subgrid scale model (SGS) in regions where the grid density is fine enough and as a RANS model in other regions. This is accomplished by replacing the distance to the wall, *d*, in the SA model by

$$\tilde{d} = \min(d, C_{DES}\Delta)$$

where $0.61 < C_{DES} < 0.78$ (typically 0.65) and Δ is an element length scale,

$$\Delta = \begin{cases} \left(VOL_{element}\right)^{1/3} & 3\text{-}D \\ \left(AREA_{element}\right)^{1/2} & 2\text{-}D \end{cases}$$

Since the value of $\tilde{d}$ controls the switching between SGS-LES and RANS, the user actually controls the behavior of the model through the size of meshes. Note that in DES the filter is of order Δ.

- **Two-layer zonal model**

In the two-layer zonal model, the whole domain is subdivided into a near-wall viscous region and a fully turbulent region. This division is determined by a wall-distance based Reynolds number defined by

$$R_y = \frac{\rho\sqrt{K}y}{\mu_0}$$

where y is the shortest distance from the location to the wall. In the fully turbulent region $\left(R_y > R_c\right)$, the standard K-ε model is employed. However, in the near-wall viscous region, a general one-equation model is employed. In this model, the turbulent viscosity μ_T and ε are computed from

$$\mu_T = \rho C_\mu K^{1/2} l_\mu$$
$$\varepsilon = K^{3/2} / l_\varepsilon$$

where the length scales are computed from

$$l_\mu = c_l y \left(1 - e^{-R_y / A_\mu}\right)$$
$$l_\varepsilon = c_l y \left(1 - e^{-R_y / A_\varepsilon}\right)$$

Note that , in the near-wall viscous region, ε is not obtained by solving the ε-equation; it is instead obtained algebraically from the above formula.

An indicator of a good mesh for the two-layer zonal model is that the nondimensional wall distance, $y^+ \approx 1$. Another indication of an ideal mesh quality is that at least 10 layers of elements are within the viscous layer. Since the parameter R_c plays a role in switching between the two models, its values are somewhat flexible. It is more convenient to specify the number of viscous layers, rather than the value of R_c. Therefore, we allow the user to input a range of $R_c \in \left[R_{c\min}, R_{c\max}\right]$, as well as a preferred number of near-wall layers $\left(L_w\right)$. The program adjusts R_c in the given range such that the number of viscosity-affected element layers is as close to L_w as possible. It is understandable that, even with the automatic adjustment, a sufficient number of elements must be placed in near-wall regions.

The two-layer zonal model is available for the FCBI-C and FCBI elements and is always associated with the turbulence K-ε model.

In practice, the empirical constants A_μ and A_ε are found to be not only physically dependent, but also "scheme-dependent". They are constants that should be calibrated and embedded into the program. In our study, the preferable values of A_μ and A_ε are found to be the same and shown in the table below.

Cases	Values of A_μ and A_ε	
	FCBI-C elements	FCBI elements
3D	105.5	100
2D Axisymmetry	52	19
2D plane	50	70

The default values of the other empirical constants are

$$c_l = 2.43, \quad R_{c\min} = 50, \quad R_{c\max} = 400, \quad L_w = 10$$

In order to obtain converged solutions at each time step, the near-wall elements are fixed after a certain number of iterations. The convergence criteria are

$$\begin{aligned} &\textit{either} \quad i \geq I_{upper} \\ &\textit{or} \quad\quad i \geq I_{lower} \textit{ and } r_e \leq \min\left\{\sigma_e \varepsilon_e, \varepsilon_{e\text{-}lower}\right\} \end{aligned}$$

where i is the iteration number, and r_e and ε_e are, respectively, the effective residual and effective tolerance. The values of the constants $I_{upper}, I_{lower}, \sigma_e$ and $\varepsilon_{e\text{-}lower}$ are currently fixed and shown in the table

	I_{lower}	I_{upper}	σ_e	$\varepsilon_{e\text{-}lower}$
FCBI-C	30	100	10	0.01
FCBI	7	20	10	0.01

- K-ε realizable turbulence model in incompressible flows

The K-ε realizable turbulence model is the same as the standard K-ε turbulence model except that the parameters c_1, c_2 and c_μ are modified to

$$c_1 = \frac{c_{1R}}{c_\mu A}$$

$$c_2 = \frac{c_{2\varepsilon}}{1+\sqrt{\nu\,\varepsilon/K^2}}$$

$$c_\mu = \frac{1}{A_0 + A_s A_*}$$

where $c_{1\varepsilon} = 1.44$, $c_{2\varepsilon} = 1.9$ are the default constants in the standard K-ε turbulence model, $A_0 = 4.04$ and

$$c_{1R} = \max\left[0.43, \frac{A}{A+5}\right]$$

$$\left(A, A_*\right) = \sqrt{2}\,\frac{K}{\varepsilon}\left(D, D_*\right)$$

$$D_* = \sqrt{D^2 + \Omega^2}$$

$$A_s = \sqrt{3}\cos\phi$$

$$\phi = \tfrac{1}{3}\cos^{-1}\left(\sqrt{6}W\right)$$

$$W = \frac{\left(\mathbf{e}\cdot\mathbf{e}^T\right)\otimes\mathbf{e}}{D^3}$$

$$\Omega^2 = \boldsymbol{\omega}\otimes\boldsymbol{\omega}$$

$$\boldsymbol{\omega} = \tfrac{1}{2}\left(\nabla\mathbf{v} - \nabla\mathbf{v}^T\right)$$

2.12 Mass transfer

- **Mass averages**

Consider *n* neutral species of density $\rho_i, (i = 1, 2, ..., n)$ that are transported in a fluid medium of density ρ_0. The mass density is defined as usual as the mass of species per unit volume. It is sometimes also called mass concentration or partial density of the mixture.

The mass of each of these species must be conserved if there is no exchange between these species and the environment. Allowing a mass creation rate m_i''' generated by other species, the mass conservation of each of the species reads

$$\frac{\partial \rho_i}{\partial t} + \nabla \bullet (\rho_i \mathbf{v}_i) = m_i'''$$

where $\mathbf{v}_i$ is the velocity of the species i. Summing up the mass conservations of all species, we obtain

$$\frac{\partial}{\partial t}\left(\sum_{i=0}^{n} \rho_i\right) + \nabla \bullet \left(\sum_{i=0}^{n} \rho_i \mathbf{v}_i\right) = \left(\sum_{i=0}^{n} m_i'''\right)$$

When there is no mass exchange between the considered media and their environment, the right-hand side of this equation becomes zero. A comparison of this equation with the general fluid conservative equation reveals two key concepts in mass transfers, namely, the bulk density and the bulk velocity of the media. They are defined respectively as

$$\rho = \sum_{i=0}^{n} \rho_i, \quad \mathbf{v} = \frac{1}{\rho} \sum_{i=0}^{n} \rho_i \mathbf{v}_i$$

According to Fick's law, the deviation of the mass flow rate is proportional to the corresponding mass concentration gradient

$$\frac{\partial \rho \phi_i}{\partial t} + \nabla \bullet (\rho \mathbf{v} \phi_i - \mathbf{D}_i \bullet \nabla \phi_i) = m_i$$

where $\mathbf{d}_i$ is the mass diffusivity, sometimes called mass diffusion coefficient. In general, $\mathbf{d}_i$ is a tensor of the second order. Combining Fick's law and the species conservation equation, we obtain the mass transfer equation

$$\frac{\partial \rho_i}{\partial t} + \nabla \bullet \left(\rho_i \mathbf{v} - \mathbf{d}_i \bullet \nabla \rho_i \right) = m_i'''$$

We may multiply this equation by an arbitrary constant, say ρ_*, to represent the same equation as

$$\frac{\partial \rho_* \rho_i}{\partial t} + \nabla \bullet \left(\rho_* \rho_i \mathbf{v} - \mathbf{d}_i \bullet \nabla \rho_* \rho_i \right) = \rho_* m_i'''$$

In this case, the solution variable and the mass creation rate become $\rho_* \rho_i$ and $\rho_* m_i'''$ respectively. This fact is important for us later to generalize our approach to mass transfer problems.

In many applications, it is more convenient to use the mass-ratios

$$\phi_i = \frac{\rho_i}{\rho}$$

as solution variables. If the bulk density can be well approximated by a constant, as it is in incompressible flows, the mass transfer equation can be represented by

$$\frac{\partial \rho \phi_i}{\partial t} + \nabla \bullet \left(\rho \mathbf{v} \phi_i - \rho \mathbf{d}_i \bullet \nabla \phi_i \right) = m_i'''$$

- **Molar averages**

Another important approach to mass transfer problems is the so-called molar concentration, particularly in gaseous mixtures. In this approach, the molar concentration $C_i = N_i / V$ is used as the solution variable, where N_i

is the number of moles of species i in the mixture and V is the volume of the mixture. Since the molecular weight M_i is constant and $\rho_i = N_i M_i / V = C_i M_i$, the mass conservation of the species *i* can be written as

$$\frac{\partial C_i}{\partial t} + \nabla \bullet \left(C_i \mathbf{v}_i \right) = \frac{m_i'''}{M_i}$$

Similarly, we can introduce the concepts of the bulk molar density and the molar averaged velocity respectively by

$$C = \sum_{i=0}^{n} C_i, \quad \hat{\mathbf{v}} = \frac{1}{C} \sum_{i=0}^{n} C_i \mathbf{v}_i$$

Applying Fick's law to the molar flux $C_i(\mathbf{v}_i - \hat{\mathbf{v}}) = -\hat{\mathbf{d}}_i \bullet \nabla C_i$ results in the molar transfer equation

$$\frac{\partial C_i}{\partial t} + \nabla \bullet \left(C_i \hat{\mathbf{v}} - \hat{\mathbf{d}}_i \bullet \nabla C_i \right) = \frac{m_i'''}{M_i}$$

Again, introducing an arbitrary constant, we obtain

$$\frac{\partial \rho_* C_i}{\partial t} + \nabla \bullet \left(\rho_* C_i \hat{\mathbf{v}} - \hat{\mathbf{d}}_i \bullet \nabla \rho_* C_i \right) = \frac{\rho_* m_i'''}{M_i}$$

in which we treat $\rho_* C_i$ as the solution variable and $\rho_* m_i''' / M_i$ as the creation rate.

Similarly, the molar-ratio $\varphi_i = C_i / C$ can also be used as a solution variable. When the molar mass of the mixture $M \left(= \rho / C = \sum_{i=0}^{n} \varphi_i M_i \right)$ and the bulk molar density *C* are considered constant, the molar transfer equation can be represented by

$$\frac{\partial \rho \varphi_i}{\partial t} + \nabla \bullet \left(\rho \varphi_i \hat{\mathbf{v}} - \rho \hat{\mathbf{d}}_i \bullet \nabla \varphi_i \right) = \frac{M m_i'''}{M_i}$$

- **"A single approach" to the mass transfer equations**

We have so far introduced the mass density ρ_i, the mass-ratio ϕ_i, the molar density C_i and the molar-ratio φ_i. Their properties are governed by the corresponding equations introduced earlier in this section. These equations fundamentally behave in the same manner and can be solved in a single formulation as follows.

Let the solution variable, diffusion coefficient and source term be denoted by ϕ_i, $\mathbf{D}_i$ and m_i, respectively. Let the transport equation that governs the solution variable be represented by

$$\frac{\partial \rho \phi_i}{\partial t} + \nabla \bullet \left(\rho \mathbf{v} \phi_i - \mathbf{D}_i \bullet \nabla \phi_i \right) = m_i$$

together with a general definition of the "bulk density" through a general formula

$$\rho = \sum_{i=0}^{n} \alpha_i \rho_i \equiv \sum_{i=0}^{n} \alpha_i \left(\rho \phi_i \right) \tag{0.12}$$

Based on the choice of solution variables, every term in the equation has a specific meaning. In particular, irrespective of the choice of the solution variable, the "bulk density" has the specific expression given above.

- The most direct choice of a solution variable is the mass-ratio. In this case, ϕ_i is the mass-ratio, $\mathbf{D}_i = \rho \mathbf{d}_i$, $m_i = m_i'''$ and the "bulk density" is the physically meaningful bulk density. The coefficients of the bulk density are then $\alpha_i = 1, \left(i = 1, 2, ..., n \right)$. The bulk density is then expressed in terms of the mass-ratio

$$\rho = \rho_0 + \rho \sum_{i=1}^{n} \phi_i = \frac{\rho_0}{1 - \sum_{i=1}^{n} \phi_i}$$

In the event that the mass-ratios are small, the problem can be well

approximated by choosing $\alpha_i = 0, (i = 1, 2, ..., n)$. In this case, the fluid equation is separated from the mass transfer equation. If the mass creations are further zeros, the mass-ratios are decoupled from each other as well. The resulting equation represents the simplest yet most popular formulation of mass transfer in incompressible flows.

- When the molar-ratio is used for the solution variable, ϕ_i represents the molar-ratio φ_i, $\mathbf{D}_i = \rho\widehat{\mathbf{d}}_i$, $m_i = Mm_i'''/M_i$ and the "bulk density" is also the physical bulk density. In order to write the bulk density in terms of molar-ratios, we follow the definitions of molar averages

$$\begin{aligned}\rho &= \rho_0 + \sum_{i=1}^{n} \rho_i \\ &= \rho_0 + CM \sum_{i=1}^{n} \frac{C_i M_i}{CM} \\ &= \rho_0 + \rho \sum_{i=1}^{n} \frac{M_i}{M} \varphi_i\end{aligned}$$

which shows that $\alpha_0 = 1$ and $\alpha_i = M_i/M \ (i = 1, 2, ..., n)$. The bulk density can then be similarly expressed in terms of the molar-ratio

$$\rho = \frac{\rho_0}{1 - \sum_{i=1}^{n} \alpha_i \phi_i}$$

As before, if the molar-ratios are small, the problem can be well approximated by choosing $\alpha_i = 0, (i = 1, 2, ..., n)$.

- When the solution variable is the mass density, ϕ_i represents the mass density ρ_i, $\mathbf{D}_i = \rho_* \mathbf{d}_i$, $m_i = \rho_* m_i'''$ and "bulk density" ρ represents any arbitrary constant ρ_*. A reasonable choice is the fluid density ρ_0. This formulation can therefore only be applied to incompressible flows. The obvious choice of the coefficients in the "bulk density" is $\alpha_0 = 1$ and $\alpha_i = 0, (i = 1, 2, ..., n)$.

- When the solution variable is the molar density, ϕ_i represents the molar density C_i, $\mathbf{D}_i = \rho_* \hat{\mathbf{d}}_i$, $m_i = \rho_* m_i''' / M_i$ and the "bulk density" ρ represents any arbitrary constant ρ_*. A reasonable choice is the fluid density ρ_0. This formulation can therefore only be applied to incompressible flows. The obvious choice of the coefficients in the "bulk density" is $\alpha_0 = 1$ and $\alpha_i = 0, (i = 1, 2, ..., n)$.

For simplicity of description of the mass transfer solutions, we will always call ϕ_i the "mass-ratio", ρ is the "bulk density", $\mathbf{D}_i$ the "mass diffusion coefficient" and m_i the "mass creation rate".

The governing equations for mass transfer in axisymmetric case are

$$\frac{\partial y \rho \phi_i}{\partial t} + \nabla \bullet \left[y \left(\rho \mathbf{v} \phi_i - \mathbf{D}_i \bullet \nabla \phi_i \right) \right] = y m_i$$

The general procedure to couple the mass transfer solutions with fluid flow solutions is as follows. We solve the mass transfer equation for $i = 1, 2, ..., n$ and compute the bulk density defined in Eq.(0.12). This density will be used in the mass transfer equation. Except for the high-speed compressible flow case, the bulk density will also be used in fluid equations. In this case, the density ρ_0 represents the original fluid density introduced in the previous sections. The bulk density introduced here is a correction of the fluid density due to the non-uniform distribution of additional species.

- **Correction to buoyancy force**

Similar to the Boussinesq approximation to the temperature-dependent density in incompressible flows, the mass-dependent density can also be simplified in many practical problems. In this case, we assume that the bulk density is independent of the mass-ratios except in the buoyancy force term, where it is extended to include the mass-ratios

$$\mathbf{f}_g^B = \rho \mathbf{g} = \rho_0 \mathbf{g} \left[1 - \beta \left(\theta - \theta_0 \right) + \sum_{i=1}^{n} \beta_i \left(\phi_i - \phi_{i0} \right) \right]$$

where β_i is a mass expansion coefficient and ϕ_{i0} is a reference mass-ratio in the expansion. In this approach, the coefficients of the bulk density must be $\alpha_0 = 1$ and $\alpha_i = 0, \left(i = 1, 2, ..., n \right)$, based on the assumption made here.

Corresponding to the mass solution variables, the mass-ratios, mass densities, molar-ratios and molar-densities, β_i =1, $1/\rho_0$, M_i/M and M_i/ρ_0 can be used, respectively.

2.13 Steady-state equations

The steady-state equations are obtained from the governing equations by removing all dynamic terms, i.e., we assume $\partial/\partial t = 0$. The steady-state analysis is to solve the steady-state equations. The solutions obtained in steady-state analyses are therefore in steady state.

It is noteworthy that a steady-state solution can be obtained by performing either a steady-state analysis or a transient analysis. For problems that have multi-solutions (for example, bifurcation problems), in general, transient analyses must be performed.

2.14 Arbitrary-Lagrangian-Eulerian formulations

In fluid dynamics, moving boundaries are frequently encountered, for example, prescribed moving walls, deformable structures, free surfaces, etc. (see the following figure for a typical example). When any part of the computational domain is deformable, the Eulerian description of fluid flows is no longer applicable and the Lagrangian description must be used. Anywhere else in the fluid domain, the fluid flow can be described in an

arbitrary coordinate system as long as it meets the coordinate requirements along the physical boundaries. Such a description is called an Arbitrary-Lagrangian-Eulerian (ALE) formulation. Obviously, the Navier-Stokes equations discussed so far need to be rewritten in the ALE system.

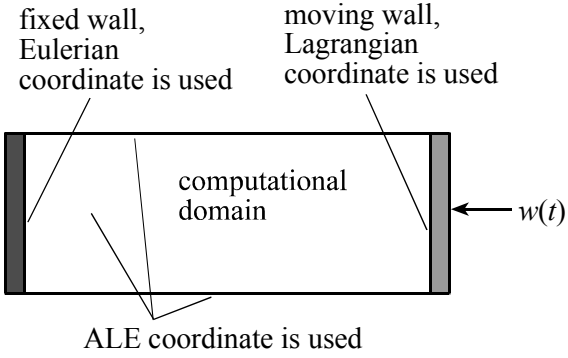


Figure 2.4 A typical example of using ALE coordinates

The purpose of choosing an ALE system is to "freeze" the computational domain. Let us consider a transformation such that the new coordinate is the sum of the initial coordinate and its displacement:

$$\mathbf{x} = \boldsymbol{\xi} + \mathbf{d}(\boldsymbol{\xi}, \tau) \equiv \mathbf{x}(\boldsymbol{\xi}, \tau)$$
$$t = \tau$$

where the moving coordinate system $(\mathbf{x}, t)$ has been transformed into a new coordinate system $(\boldsymbol{\xi}, \tau)$. Note that, to give a more precise explanation, we have used τ to represent the time in the new coordinate system. The vector $\mathbf{d}(\boldsymbol{\xi}, \tau)$ is used to deal with the arbitrarily moving coordinates.

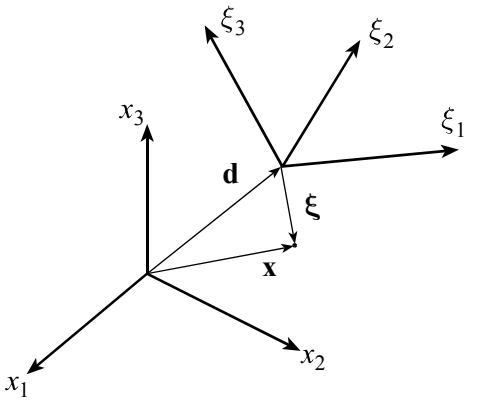


Figure 2.5 Definition of the ALE coordinate system

The time derivative of an arbitrary, $f(\mathbf{x},t) = f(\boldsymbol{\xi} + \mathbf{d}(\boldsymbol{\xi},\tau),\tau)$, is

$$\frac{\partial f}{\partial \tau} = \frac{\partial f}{\partial t} + \frac{\partial \mathbf{x}}{\partial \tau} \bullet \frac{\partial f}{\partial \mathbf{x}}$$

Using $\mathbf{w}$ ($\equiv \partial \mathbf{x}/\partial \tau = \partial \mathbf{d}/\partial \tau$) to represent the moving coordinate velocity, we have

$$\frac{\partial f}{\partial t} = \frac{\partial f}{\partial \tau} - \frac{\partial \mathbf{x}}{\partial \tau} \bullet \frac{\partial f}{\partial \mathbf{x}} = \frac{\partial f}{\partial \tau} - \mathbf{w} \bullet \nabla f \qquad (0.13)$$

When this equation is applied to time derivatives in the Navier-Stokes equations that are expressed in a purely Eulerian coordinate system, the differential forms of ALE equations are obtained. For example, in the ALE system, the nonconservative continuity equation becomes,

$$\frac{\partial \rho}{\partial \tau} + (\mathbf{v} - \mathbf{w}) \bullet \nabla \rho + \rho \nabla \bullet \mathbf{v} = 0$$

Hence, the only difference between an ALE formulation and the Eulerian formulation is that the relative velocity replaces the convective velocity. Furthermore, the case $\mathbf{w} = \mathbf{0}$ corresponds to a purely Eulerian description, while $\mathbf{w} = \mathbf{v}$ corresponds to a purely Lagrangian description.

When Eq.(0.13) is directly used, however, only the nonconservative forms of Navier-Stokes equations can be obtained. The conservative ALE Navier-Stokes equations can only be expressed in their integral forms. In general, a conservative form of the differential equation

$$\frac{\partial \mathbf{f}}{\partial t} + \nabla \bullet \mathbf{B} = \mathbf{R}$$

in a fixed coordinate system will be transformed into an integral form in an ALE coordinate system

$$\frac{\partial}{\partial \tau}\int_V \mathbf{f} dV + \oint_S (-\mathbf{w}\mathbf{f} + \mathbf{B}) \bullet d\mathbf{S} = \int_V \mathbf{R} dV$$

\where $V(t)$ is an arbitrary material volume enclosed by its surface $S(t)$ and $\mathbf{S}$ is the surface vector of $S(t)$ in the outward direction. The quantities $\mathbf{f}$, $\mathbf{B}$ and $\mathbf{R}$ are tensors such that $\mathbf{B}$ is one order higher than $\mathbf{f}$ and $\mathbf{R}$. In particular, the volume can be any control volume if finite volume methods are used. We will frequently refer to this equation when finite volume methods are discussed.

Using this equation, all the conservative forms of equations introduced in the previous sections can be directly transformed into their corresponding integral forms of conservation. For example, the compact form of the conservative equation can be written in the following integral form, in an arbitrarily moving system,

$$\frac{\partial}{\partial \tau}\int_V \mathbf{U} dV + \oint_S [\mathbf{F}(\Delta \mathbf{v}, \mathbf{U}) + \mathbf{G}] \bullet d\mathbf{S} = \int_V \mathbf{C} dV$$

where $\Delta \mathbf{v} = \mathbf{v} - \mathbf{w}$ is the relative velocity. Furthermore, the Jacobian of $\mathbf{F}_n(\Delta \mathbf{v}, \mathbf{U})$ has the same eigenvectors as defined in Eqs.(0.9) and (0.10). Its eigenvalues, however, have been modified to $\mathbf{D}(\Delta u, c)$, where $\Delta u = u - \mathbf{n} \bullet \mathbf{w} \equiv u - w_n$.

2.15 System units and nondimensional forms

The Navier-Stokes equations and associated conditions and material data can be used in any system of units (S.I. units, USCS units, etc.) as long as all units are consistent. In particular, dimensionless "units" can be used. To define nondimensional units, characteristic values of all the variables entering the Navier-Stokes equations must be constructed from a predetermined set of independent reference quantities and scales. They are listed in the Nomenclature.

All other variables can then be derived from these basic quantities and scales. For example, using the tilde to indicate dimensionless quantities, we have

$$\tilde{\mathbf{x}} = \frac{\mathbf{x} - \mathbf{x}_r^*}{L^*}, \qquad \tilde{t} = \frac{tV^*}{L^*}, \qquad \tilde{\rho} = \frac{\rho}{D^*},$$

$$\tilde{\mathbf{v}} = \frac{\mathbf{v}}{V^*}, \qquad \tilde{p} = \frac{p}{D^* V^{*2}}, \quad \tilde{\mu} = \frac{\mu}{D^* V^* L^*},$$

$$\tilde{\lambda} = \frac{\lambda}{D^* V^* L^*}, \quad \tilde{\mathbf{f}}^B = \frac{\mathbf{f}^B L^*}{D^* V^{*2}}$$

The nonconservative momentum equations become, using nondimensional quantities,

$$\tilde{\rho}\frac{\partial \tilde{\mathbf{v}}}{\partial \tilde{t}} + \tilde{\rho}\tilde{\mathbf{v}} \bullet \tilde{\nabla}\tilde{\mathbf{v}} - \tilde{\nabla} \bullet \tilde{\boldsymbol{\tau}} = \tilde{\mathbf{f}}^B$$

where

$$\tilde{\boldsymbol{\tau}} = \left(-\tilde{p} + \tilde{\lambda}\tilde{\nabla} \bullet \tilde{\mathbf{v}}\right)\mathbf{I} + 2\tilde{\mu}\tilde{\mathbf{e}}$$

$$\tilde{\mathbf{e}} = \frac{1}{2}\left(\tilde{\nabla}\tilde{\mathbf{v}} + \tilde{\nabla}\tilde{\mathbf{v}}^T\right)$$

It is noted that material data now are $\tilde{\mu}$, $\tilde{\rho}$, etc.

For compressible flows, since the temperature is strongly coupled to the other fluid variables, we use

$$T^* = \frac{V^{*2}}{C^*}, \quad \theta_r^* = 0$$

In natural convections, since there is no free stream, a proper velocity scale can be

$$V^* = \frac{\mu^*}{D^* L^*}$$

where μ^* is a characteristic viscosity.

All parameters, in dimensional and nondimensional forms, are listed in the Nomenclature section of this manual. The following well-known dimensionless parameters are listed next since they are useful indicators of flow patterns.

- **Reynolds** number, defined as

$$\mathrm{Re} = \frac{D^* V^* L^*}{\mu} = \frac{1}{\tilde{\mu}}$$

is the ratio of the inertia force to the friction force, which measures the intensity of turbulence and the boundary layer thickness. Typical transition Reynolds numbers to turbulent flows, for internal and external flows respectively, are of the order 10^3 and 10^6. The typical fluid boundary layer thickness is of the order $\mathrm{Re}^{-1/2} L^*$.

- **Prandtl** number, defined as

$$\mathrm{Pr} = \frac{\mu C^*}{k} = \frac{\tilde{\mu}}{\tilde{k}}$$

is the ratio of the diffusivity of momentum to the diffusivity of heat. Note that the Prandtl number depends only on fluid properties. Typical Pr values

are, under normal conditions, 0.01, 0.7, 10 and 1000 for mercury, gas, water and oil, respectively.

- **Rayleigh** number, defined as

$$\mathrm{Ra} = \frac{D^{*2}\|\mathbf{g}\|L^{*3}\beta T^{*}C^{*}}{\mu k} = \frac{\|\tilde{\mathbf{g}}\|\tilde{\beta}}{\tilde{\mu}\tilde{k}} = \mathrm{Pr}\,\mathrm{Gr}$$

is the ratio of the energy generated by the buoyancy force to the energy dissipated by thermal diffusion and fluid viscosity. Ra is an important parameter in natural convection problems to measure the intensity of flows and the thickness of the thermal boundary layers. When the inertia force is negligible in the momentum equation, the typical thermal boundary layer thickness is of the order $\mathrm{Ra}^{-1/4}L^{*}$. In another limit, when the viscous force is negligible, the typical thermal boundary layer thickness is of the order $\mathrm{Bo}^{-1/4}L^{*}$, where $\mathrm{Bo} = \mathrm{RaPr}$ is the Boussinesq number.

- **Mach** number, defined as

$$\mathrm{M} = \frac{\|\mathbf{v}\|}{c}$$

is the ratio of the fluid velocity to the speed of sound . The Mach number is an important parameter in determining the regime of fluid flows. It is usually acceptable that flows can be treated as incompressible when the Mach number is smaller than 0.3. Once the Mach number exceeds one, flows become supersonic and many kinds of phenomena related to shocks may occur.

- **Boussinesq** number, defined as

$$\mathrm{Bo} = \frac{D^{*2}\|\mathbf{g}\|L^{*3}\beta T^{*}C^{*2}}{k^{2}} = \frac{\|\tilde{\mathbf{g}}\|\tilde{\beta}}{\tilde{k}^{2}} = \mathrm{RaPr}$$

is the ratio of the energy generated by the buoyancy force to the energy dissipated by thermal diffusion. As mentioned earlier, when the contribution of the flow viscosity is negligible, the Boussinesq number measures the thermal boundary layer thickness.

- **Grashof** number, defined as

$$\mathrm{Gr} = \frac{D^{*2}\|\mathbf{g}\|L^{*3}\beta T^{*}}{\mu^{2}} = \frac{\|\tilde{\mathbf{g}}\|\tilde{\beta}}{\tilde{\mu}^{2}} = \frac{\mathrm{Ra}}{\mathrm{Pr}}$$

is the ratio of the buoyancy force to the viscous force. When Pr < 1, the Grashof number is an indicator of the thickness of fluid shear viscous layer within the thermal boundary layer. The typical thickness of the shear layer is of the order $\mathrm{Gr}^{-1/4}L^{*}$.

- **Peclet** number, defined as

$$\mathrm{Pe} = \frac{D^{*}V^{*}L^{*}C^{*}}{k} = \frac{1}{\tilde{k}}$$

is the ratio of the heat transferred by thermal convection to the heat transferred by conduction. The Peclet number is an important parameter in forced convection problems. The thickness of the thermal boundary layer is of the order $\mathrm{Pe}^{-1/2}L^{*}$.

- **Eckert** number, defined as

$$\mathrm{Ec} = \frac{V^{*2}}{C^{*}T^{*}}$$

is the ratio of the kinetic energy to the internal energy. In compressible flows, Eckert number is of the same order as the Mach number. For a perfect gas,

$$\mathrm{Ec} = (\gamma - 1)\mathrm{M}^{2}\tilde{C}_{p}$$

- **Schmidt** number, defined as

$$\mathrm{Sc} = \frac{\mu}{\|\mathbf{D}\|} = \frac{\tilde{\mu}}{\|\tilde{\mathbf{D}}\|}$$

is the ratio of the diffusivity of fluids to the diffusivity of mass-ratios. Schmidt number is the analog of the Prandtl number between the fluid and the mass transfer. For most common liquids, Sc is two or more orders of magnitude greater than Pr. For gases, Sc is typically of the same order of magnitude as Pr.

- **Lewis** number, defined as

$$\mathrm{Le} = \frac{\|\mathbf{D}\| C^*}{k} = \frac{\|\tilde{\mathbf{D}}\|}{\tilde{k}} = \frac{\mathrm{Pr}}{\mathrm{Sc}}$$

is the ratio of the diffusivity of the mass-ratio to the diffusivity of heat. It is typically of the order 10^{-2} or 10^{-3} for most common liquids and order one for gases.

- **Capillary** number, defined as

$$\mathrm{Ca} = \frac{\mu V^*}{\sigma} = \frac{\tilde{\mu}}{\tilde{\sigma}}$$

is the ratio of the viscous forces to the surface tension forces. The smaller the Capillary number, the more important the effect of surface tension.

- **Brinkman** number, defined as

$$\mathrm{Br} = \frac{\mu V^{*2}}{kT^*} = \mathrm{Pr}\,\mathrm{Ec}$$

is the ratio of the energy dissipated by the fluid viscosity to the energy dissipated by thermal conduction.

Chapter 3 Formulation of incompressible, slightly compressible and low-speed compressible flows

In this chapter, we introduce our formulation for incompressible, slightly compressible and low-speed compressible flows. Heat transfer can be included in incompressible and slightly compressible flows. In low-speed compressible flows, the temperature must be included.

The fluid flow models introduced here can be coupled with ADINA solid models in fluid-structure interaction analyses. They can also be associated with mass transfer analyses. Porous media models can be employed as additional element groups.

Solving a fluid flow problem involves solving the governing equations associated with the input of physical fluid material data, well-posed boundary conditions, initial conditions and the use of appropriate numerical methods. These issues are discussed in this chapter.

ref. Bathe, K.J. and Zhang, H., "A flow-condition-based interpolation finite element procedure for incompressible fluid flows," *Computers & Structures*, Vol. 80, pp. 1267-1277, 2002.

ref. Bathe, K.J., Zhang, H., and Ji, S., "Finite Element Analysis of Fluid Flows fully Coupled with Structural Interactions," *Computers & Structures*, Vol. 72, pp. 1-16, 1999.

ref. Bathe, K.J., Zhang, H., and Zhang, X., "Some Advances in the Analysis of fluid flows," *Computers & Structures*, Vol. 64, pp. 909-930, 1997.

ref. Bathe, K.J., Zhang, H., and Wang, M.H., "Finite Element Analysis of Incompressible and Compressible fluid flows with Free Surfaces and Structural Interactions," *Computers & Structures*, Vol. 56, pp. 193-214, 1995.

ref. Bathe, K.J., Walczak, J., and Zhang, H., "Some Recent Advances for Practical Finite Element Analysis," *Computers & Structures*, Vol. 47, pp. 511-521, 1993.

The following table presents a quick overview of the capabilities that are available for incompressible, slightly compressible and low-speed compressible flows.

Table 3-1 Functionalities of incompressible, slightly compressible and low-speed compressible flow models

Category	**Functionality**
Coupled models	ADINA solid model
	porous medium element groups
	solid element groups
	heat transfer (always coupled in compressible flows)
	mass transfer
	electro-static and steady current conduction analyses
	volume of fraction
	liquid-vapor phase change
Computational domains	2D planar, 2D axisymmetric and 3D
Analyses	steady-state and transient
Galerkin Elements	2D: 3-node linear element and 6/9-node bilinear elements
	3D: 4-node linear element and 27-node bilinear elements
FCBI and FCBI-C elements	2D: 3/4-node linear elements
	3D: 4/5/6/8-node linear elements
Material models	constant
	time-dependent
	power-law
	Carreau
	temperature-dependent
	temperature-dependent power-law
	second order
	large-eddy-simulation
	ASME steam table
	Pressure-temperature-dependent
	user-supplied

Boundary conditions Note: The column furthest to the right represents the solution variables or equations that are immediately affected by the specified condition. The solution variable **d** indicates a moving boundary condition.	prescribed velocities	**v**
	zero velocities	
	prescribed pressure	p
	zero pressure	
	prescribed rotational velocity	**v**
	concentrated force load	
	distributed normal-traction load	
	field centrifugal load	
	fixed wall	
	uniform flow	
	angular velocity	
	moving wall	**v, d**
	fluid-structure interface	
	free surface	**D**
	fluid-fluid interface	
	phase-change	
	gap	all
	prescribed temperature	θ
	zero temperature	
	concentrated heat flow load	
	distributed heat flux load	
	convection	
	radiation	
	specular radiation	
	thermal resistance	
	user-supplied	all
Initial conditions Note: Zero pressure or temperature cannot be specified in compressible models.	zero (default conditions)	
	specified	
	mapped from other solutions	
	restart run	
Solvers for linearized equations	Gauss elimination method (COLSOL)	
	sparse solver	
	iterative methods (RPBCG, RPGMRES, and AMG solvers)	
Other capabilities	automatic nondimensionalization procedure	
	automatic time-stepping CFL option	
	automatic time-stepping ATS option	
	skew systems	
	constraint condition	
	conjugate heat transfer	
	element birth-death option	
	pressure datum	
	include/exclude hydrostatic pressure	
	physical or mathematical formulations for incompressible flows	
	with or without dissipations	

3.1 Governing equations

The governing equations used in our formulations are in general given in Chapter 2, in various forms, corresponding to different cases. Irrespective of the solution variables used, the final solutions for visualization purpose are always primitive variables $(p, \mathbf{v}, \theta, ...)$. The ALE formulation described in Section 2.14 is used for moving mesh problems.

It is worth noting that the heat transfer equation is used as default. For FCBIC element and low-speed compressible fluid, user can choose either the heat transfer equation (as default) or the conservative form of the energy equation that includes the total energy and the state equation of gas. In practice, heat transfer equation is robust, but is not always accurate. An example where the heat transfer equation is not accurate is quasi-static polytropic compression and expansion. On the other hand, total energy equation is more accurate, but could be numerically less stable at higher flow velocities.

It is worth noting that the heat transfer equation is used as default. For FCBIC element and low-speed compressible fluid, user can choose either the heat transfer equation (as default) or the conservative form of the energy equation that includes the total energy and the state equation of gas. In practice, heat transfer equation is robust, but is not always accurate. An example where the heat transfer equation is not accurate is quasi-static polytropic compression and expansion. On the other hand, total energy equation is more accurate, but could be numerically less stable at higher flow velocities.

3.2 Numerical method

3.2.1 Time integration

In steady-state analyses, the transient terms in the governing equations are removed and therefore there is no time integration involved. In this case, time will only be used in time-dependent materials, loads and boundary conditions. The time integration method discussed here is only for transient analyses.

Assume that the solution has been obtained at time t and the next solution is to be calculated at time $t+\Delta t$, where Δt is the time step size. The initial condition defines the solution at time 0. The algorithm used to obtain the solution at time $t+\Delta t$ is the basic procedure to obtain, successively, the solutions at all required times.

We use two implicit time integration methods: the Euler α- method of the first order Euler α-method and the ADINA composite scheme of the second order.

With an Euler method, the equation $\partial u/\partial t = f(u)$ is computed by

$$ {}^{t+\Delta t}u = {}^{t}u + \Delta t f\left({}^{t+\alpha\Delta t}u\right) $$

where ${}^{t+\alpha\Delta t}u = (1-\alpha)\,{}^{t}u + \alpha\,{}^{t+\Delta t}u$. The Euler method is a scheme of the first order accuracy and is unconditional L- stable if $\frac{1}{2} < \alpha \le 1$. Note that although it gives an accuracy of the second order in time, the trapezoidal rule ($\alpha = \frac{1}{2}$) is numerically unstable unless the velocity is extremely small. The default first order scheme is the Euler backward method ($\alpha = 1$).

In the ADINA composite method, the solution at time $t+\Delta t$ is obtained in two consecutive sub-time-steps

$$ {}^{t+\gamma\Delta t}u = {}^{t}u + \gamma\Delta t f\left({}^{t+\frac{1}{2}\gamma\Delta t}u\right) $$

$$ {}^{t+\Delta t}u = {}^{t+\beta\gamma\Delta t}u + (1-\alpha)\Delta t f\left({}^{t+\Delta t}u\right) $$

where ${}^{t+\beta\gamma\Delta t}u = (1-\beta)\,{}^{t}u + \beta\,{}^{t+\gamma\Delta t}u$, $\gamma = 2 - 1/\alpha$ and $\beta = \alpha^2/(2\alpha - 1)$. With the choice of $\frac{1}{2} < \alpha < 1$, the method has second order accuracy and is unconditionally L- stable. The default is $\alpha = 1/\sqrt{2}$, which was proven to give minimum truncation error for linear systems. Although the computational cost is doubled per time step, the composite scheme provides solutions of better accuracy and may have less CPU time overall since fewer number of time steps can be used.

3.2.2 Discretized equations

The finite element method is used to discretize the governing equations. The finite element equations are obtained by establishing a weak form of the governing equations using the Galerkin procedure. The continuity equation, the momentum equation and the energy equation are weighted with the virtual quantities of pressure, velocities and temperature. The governing equations are integrated over the computational domain V. The divergence theorem is used to lower the order of the derivatives of the stress and the heat flux, resulting in the expressions of natural boundary conditions. These conditions are part of the usual boundary conditions discussed later in this chapter.

Our variational forms of governing equations depend on the flow models, frames of coordinate systems and geometry dimensions. As an example, the conservative forms of the three-dimensional equations can be integrated and expressed in a compact form as

$$\int_V \left(h^f G^f + \mathbf{Q}^f \bullet \nabla h^f \right) dV = \oint h^f \mathbf{Q}^f \bullet d\mathbf{S} \qquad (0.14)$$

where f represents p, $\mathbf{v}$ and θ for the continuity equation, momentum equations and energy equation, respectively; h^p, $h^{\mathbf{v}}$ and h^θ are virtual quantities of pressure, velocity and temperature, and

$$G^p = \frac{\partial \rho}{\partial t} + \nabla \bullet (\rho \mathbf{v})$$

$$\mathbf{G}^{\mathbf{v}} = \frac{\partial \rho \mathbf{v}}{\partial t} + \nabla \bullet (\rho \mathbf{v}\mathbf{v}) - \mathbf{f}^B$$

$$G^\theta = \frac{\partial \rho E}{\partial t} + \nabla \bullet (\rho \mathbf{v} E - \boldsymbol{\tau} \bullet \mathbf{v}) - \mathbf{f}^B \bullet \mathbf{v} - q^B$$

$$\mathbf{Q}^p = \mathbf{0}$$

$$\mathbf{Q}^{\mathbf{v}} = \boldsymbol{\tau}$$

$$\mathbf{Q}^\theta = -\mathbf{q}$$

In particular, if h^f are step functions, the Galerkin method becomes the finite volume method (according to conventional definition). For these elements, a set of Flow-Condition-Based-Interpolation (FCBI) functions

has been used (see Chapter 10 for details). When a degree of freedom is defined at the element center, the FCBI element becomes FCBI-C element. The FCBI and FCBI-C elements can be 2D 4-node quadrilateral and 3-node triangle, 3D 8-node brick, 4-node tetrahedron, 5-node pyramid and 6-node prism. All FCBI and FCBI-C elements of the same dimensions can be mixed in one model problem.

3.2.3 Upwinding techniques

ref. KJB Sections 4.4 and 4.5

The finite element formulation of incompressible fluid flows follows the standard Galerkin procedure. Since the elements associated with this procedure satisfy the inf-sup condition, excellent results may be obtained for low Reynolds/Peclet number flows. When Reynolds/Peclet numbers become larger, the scheme becomes "less stable" and eventually unstable when certain critical conditions are violated. These conditions may be met far before the flows become physically turbulent. Therefore, such instability originates solely from the numerical methods that are employed. The numerical instability cannot be overcome by simply employing a turbulence model, since a numerically unstable method obviously cannot be applied to turbulence models as well.

ref. KJB Section 7.4.3

In order to understand the difficulties of high Reynolds/Peclet number flows, let us consider a simple example of the one-dimensional heat transfer equation expressed in nondimensional form:

$$\mathrm{Pe}\,\theta'(x) - \theta''(x) = 0$$
$$\theta(0) = 0, \quad \theta(1) = 1$$

where Pe $\left(= \rho C_p \|\mathbf{v}\| L/k\right)$ is the Peclet number and L is the length of the model problem. The exact solution of the problem is

$$\theta = \frac{e^{x\mathrm{Pe}} - 1}{e^{\mathrm{Pe}} - 1} \tag{0.15}$$

Let us use N linear elements of equal size $\Delta x = 1/N$ and apply the standard Galerkin method to the model equation. The finite element

equations that govern the variables defined at nodal points $x_i = i\Delta x, \left(i = 0,1,2,...N\right)$ are

$$-\left(1+\frac{P}{2}\right)\theta_{i-1} + 2\theta_i - \left(1-\frac{P}{2}\right)\theta_{i+1} = 0$$

where P $(= \mathrm{Pe}\Delta x)$ is called the cell Peclet number.

The exact solution (which is the numerical solution) of the discretized equation can be obtained

$$\theta_i = \frac{a^i - 1}{a^N - 1}$$

where $a = \left(1 + P/2\right)/\left(1 - P/2\right)$. It is noted that 2 is a critical value of P, at which a becomes infinity. In case $P > 2$ ($a < 0$), a^i is positive for even numbers of point indices and negative for the odd numbers. This indicates that the numerical solution oscillates when the cell Peclet number is larger than 2. It is also noted that when $P > 2$, the diagonal dominance of the matrix is no longer seen.

In the standard finite element procedure, the shape functions that are employed are "isotropic", i.e., they have equal gradients in all directions. They are good for low Reynolds/Peclet number flows, in which the equation system is close to being symmetric. For high Reynolds/Peclet number flows, the nonsymmetrical property of the underlying continuum mechanics in the equation system is strong. Such a non-uniform property cannot be well-represented by elements of uniform shape functions.

Upwinding techniques are introduced to overcome the above difficulties. The fundamental principle of upwindings is to modify the solution procedure such that the final discretized equations can more accurately represent the physical situation.

An obvious approach to a good upwinding scheme is to modify the value of a such that the numerical solution exactly equals the analytical solution e^P. We can achieve this result by modifying the heat conductivity coefficient such that

$$k\left(1+\beta\right) \rightarrow k$$

where β is an expected correction factor as a result of the upwinding. Substituting it into Eq.(0.15) and solving for β

$$\beta = \frac{P}{e^P - 1} + \frac{P}{2} - 1 \qquad (0.16)$$

Note that β is zero when P is zero, and approaches $P/2$ when P becomes large.

According to this approach, the best solution (the analytical solution) can be obtained numerically, if the standard Galerkin method is applied to the governing equation with the modified *k*.

In general, fluid flows and governing equations are nonlinear and the sizes of elements employed are non-uniform. The modification is then based on the cell Peclet number evaluated in each element.

In multidimensional domains, the construction of a good upwinding scheme is more challenging. In finite element formulations, various formulations have been proposed. In our approach to triangular and tetrahedral elements, all the integrations are evaluated analytically. Upwinding is then added based on the analytical solutions in the flow direction.

In the finite volume approach of upwinding, on the other hand, the cell Reynolds/Peclet numbers are evaluated based on the definitions of control volumes and used to determine the upwinding.

Both approaches are used to modify our standard Galerkin finite element methods. We remark that the two approaches are identical in one-dimensional solutions with constant Reynolds/Peclet numbers.

3.2.4 Solution method for nonlinear finite element equations

The finite element or finite volume equations form a coupled nonlinear system of all solution variables. The methods used for solving nonlinear equations and the linear equation solvers are described in Chapter 11.

The global matrix and the right-hand side vector are assembled using the matrices and the right-hand sides computed within each element. They also include modifications and/or additions from boundary elements that describe the specified boundary conditions.

When a zero variable condition is applied, the solution corresponding to that equation is removed from the governing equations and this procedure reduces degrees of freedom. For example, if a zero pressure is applied to node number 3, the continuity equation defined at node 3 is removed and, consequently, the number of equations is reduced.

When a prescribed variable is applied, the equation corresponding to that variable is modified as follows. If variable x_i is prescribed as $\overline{x}_i(t)$, the i-th equation in the linearized system is modified to

$$A_{ij} = \delta_{ij}, \quad B_i = \overline{x}_i(t) - x_i$$

which is equivalent to the equation

$$\Delta x_i = \overline{x}_i(t) - x_i$$

Concentrated force terms are directly added as additional terms to the right-hand side of the equations. For example, if a force $\overline{F}(t)$ is applied to a velocity component corresponding to equation i, the i-th component of the right-hand side vector is modified as

$$B_i + \overline{F}(t) \rightarrow B_i$$

Similarly, a concentrated heat flow load will be directly added to the right-hand side of the corresponding energy equation.

When a distributed normal-traction load is applied, the nodal force is computed by integrating the distributed normal-traction load, and then is added to the equation as for a concentrated force. A distributed heat flux is similarly integrated to a nodal heat flow and then added to the corresponding energy equation.

Centrifugal loads are directly formulated in the governing equation and assembled.

Unlike the treatment of usual boundary conditions, special boundary conditions are discretized within boundary elements. The boundary sides of the field elements form the boundary elements. Similar to regular elements, the coefficient matrices and the right-hand sides of boundary elements are formulated from the applied boundary condition and assembled into the global system. A boundary element is treated after the

attached field element equations are formulated and before they are assembled into the global system. The field element equations may or may not be then modified depending on the type of the special boundary condition.

For example, a radiation boundary condition results in a boundary element equation that is directly formulated in the boundary element and assembled into the global energy conservation equation. It is clear that both the matrix and the right-hand side will receive contributions from this condition.

As another example, when a slip wall condition is applied, the normal component of the momentum field element equation is replaced by the boundary condition equation. The tangential component of the finite element equation will not be changed by this condition.

Special boundary conditions can be more complicated than the two examples here. In specular radiation boundary conditions, for example, the concentrated heat flow includes the variable G that cannot be computed until the radiosity solution R is obtained. Therefore, prior to the computation of the heat flux, the radiosity equations must be solved. In order to form this equation, a ray tracing procedure is performed to setup the view factor matrix. Furthermore, these equations also depend on the boundary temperature. Thus, the two systems are actually coupled. The global iteration procedure must then be applied to the radiosity equations. Unlike the sparse matrix of the fluid equation system, the matrix of the radiosity equation is full. For most problems associated with specular boundary conditions, solving the radiosity equations is far more expensive, in terms of computational speed and memory usage, than solving the fluid equation system.

3.3 Elements

In Galerkin formulation, 2D 3-node and 3D 4-node linear elements and 2D 9/6-node and 3D 27-node bilinear elements are applicable. The upwinding technique is used in linear elements and, therefore, they are suitable for high and low Reynolds/Peclet number flows. Bilinear elements are suitable for low Reynolds/Peclet number flows.

All FCBI and FCBI-C elements are linear elements. They include 3-node triangle and 4-node quadrilateral for 2D domains; 4-node tetrahedron, 5-node pyramid, 6-node prism and 8-node brick for 3D domains. Elements of the same dimensions can be mixed in one problem.

FCBI-C elements are only associated with the Segregated method and other elements are only associated with the Newton-Raphson method. An option of the second-order in space is available for FCBI-C elements. Note that solution accuracy does depend on the element quality. In general, taking 2D 4-node element as an example, the element should be aligned along the flow direction as much as possible, and the element size variation should be as smooth as possible.

3.4 Boundary conditions

3.4.1 General descriptions

Using proper boundary conditions is essential to successfully solving fluid flow problems. The required boundary conditions depend on the type of flows or, equivalently, on the governing equations used and the associated computational domain.

Fluid velocity and pressure are always strongly coupled and must be treated like one variable as far as boundary conditions are concerned. On the other hand, the fluid flow might not be directly coupled to the temperature (although the flow may be coupled to the temperature field through temperature-dependent materials and the buoyancy force in the Boussinesq approximation). Therefore, as a general rule, temperature must be considered separately when applying boundary conditions.

A typical fluid domain and associated boundary conditions are shown in the following figure, where the whole boundary S of the computational domain has been divided into $\overline{S_1}$ (where velocities are prescribed) and $\overline{S_2}$ (where a distributed normal-traction load is applied). The S_i here represents the interior part of that part of the boundary, while the over bar indicates that it contains the boundary of S_i as well. Therefore, the common part of S_1 and S_2 is the empty geometric set, while $\overline{S_1}$ and $\overline{S_2}$

share their common frontier where both conditions are applied. This boundary partition can be expressed as

$$\overline{S}_1 \bigcup \overline{S}_2 = \overline{S} = S, \quad S_1 \bigcap S_2 = \varnothing$$

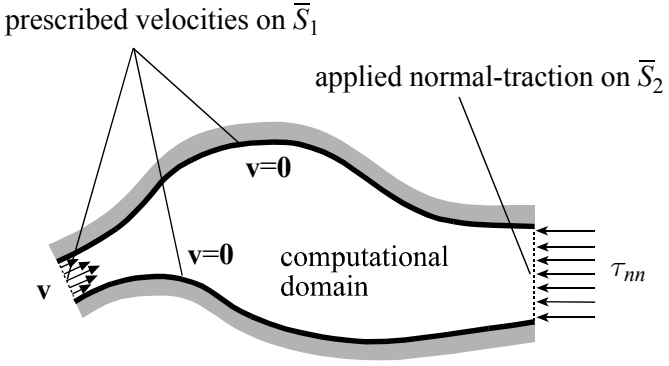


Figure 3.1 A typical boundary condition set for incompressible, slightly compressible and low-speed compressible flows

From this set of conditions, we observe that:

- All parts of the boundary have been assigned one and only one fluid boundary condition (except on the interface of the sub-boundaries, where two conditions meet).
- For steady-state analyses, at least one velocity condition and one pressure condition must be specified (the distributed normal-traction load acts as a prescribed pressure condition).

These observations are fundamental guidelines for specifying fluid boundary conditions in most fluid flow problems.

A similar boundary condition set for temperature is shown in the following figure, where the whole boundary S of the computational domain has been divided into $\overline{S}_3$ (where a temperature is prescribed) and $\overline{S}_4$ (where heat fluxes are applied). This boundary partition can be expressed as:

$$\overline{S}_3 \bigcup \overline{S}_4 = \overline{S} = S, \quad S_3 \bigcap S_4 = \varnothing$$

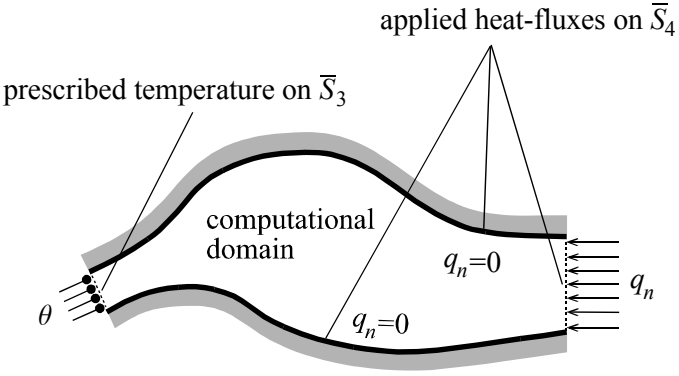


Figure 3.2 A typical boundary condition set for temperature

From this set of conditions, we have similar observations:

- All parts of the boundary have been assigned one and only one temperature boundary condition (except on the interface of the sub-boundaries, where two conditions meet).
- For steady-state analyses, at least one prescribed temperature condition must be specified. Notice that a convection condition or a radiation condition acts as a prescribed temperature condition.

Variables directly imposed on the solution are referred to as usual boundary conditions. They are applied to the governing equations in a simple manner. These conditions are either prescribed conditions or natural boundary conditions. A prescribed condition forces the solution variable to be equal to the prescribed value and the corresponding equation is replaced. A natural boundary condition acts as a force (to the momentum equation) or a heat flux (to the energy equation), and is added to the right-hand side of the corresponding equations.

Other boundary conditions may be more complicated and strongly depend on the solution variables. We call them special boundary conditions. Unlike "usual boundary conditions", the "special boundary conditions" containing unknown variables are discretized and assembled

into the global system of equations. The conditions are usually, although not necessarily, nonlinear and complex.

Unless noted otherwise, special boundary conditions can only be applied to "geometries" of one dimension lower than the computational domain. More specifically, in 2D models, the "geometries" are lines or nodes that can be connected to lines; and in 3D models the "geometries" are surfaces or nodes that can be connected to surfaces. These "geometries" may be located on the boundary or inside the domain, depending on the type of boundary conditions.

Most boundary conditions impose only constraints on the fluid variables (velocity, pressure, temperature, etc.). Some other conditions impose additional constraints on the boundary positions or boundary displacements. These conditions are called kinematic conditions or moving boundary conditions. When such a condition is applied, the boundary nodal displacements are either computed or specified. The nodal positions inside the fluid domain are adjusted in an "arbitrary" way to ensure a good mesh quality. In this case, an arbitrary Lagrangian-Eulerian (ALE) coordinate system is used in the governing equations. It is understood then that additional variables (displacements) add additional difficulties and computer costs in the solution procedure.

Unless explained otherwise, we have used the following notation in the discussion of boundary conditions. A time-dependent value, say $\overline{b}(t)$ is defined as

$$\overline{b}(t) = \overline{b}^{\,p} f_b(t)$$

by a prescribed constant value $\overline{b}^{\,p}$ (which has a unit that corresponds to the prescribed variable) and an associated time function $f_b(t)$ (which has no units).

3.4.2 Usual boundary conditions for fluid

3.4.2.1 Prescribed velocity

In this condition, a time-dependent velocity is directly prescribed

$$v_i = \overline{v}_i(t)$$

and applied to boundaries. The x_i -momentum equations at the boundary nodes are then replaced by this condition.

The prescribed velocity condition is usually applied to inlet boundaries where velocities are known. It can also be applied to fixed solid walls. When the Eulerian coordinate system is used on a moving wall, this condition can also be applied. In this case, the prescribed fluid velocity equals the velocity of the moving wall, while the displacement is zero. If non-zero displacements of the moving wall must be modeled, the moving wall conditions must be applied.

There is also a time birth and death option associated with the prescribed velocity. If only a birth option is used, the prescribed condition is inactive until the solution time is larger than the birth time. If only the death option is used, the prescribed condition is active until the solution time is larger than the death time. If both birth and death options are used, the prescribed condition is only active between the birth time and the death time. It is then understood that the death time must be larger than the birth time. While the prescribed condition is inactive, the velocity degree of freedom is free.

3.4.2.2 Zero velocity

Physically, this condition is equivalent to a prescribed zero velocity. However, the zero velocity condition removes the velocity degree of freedom. Hence the birth and death options are not applicable.

3.4.2.3 Prescribed pressure

In this condition, a time-dependent pressure is directly prescribed

$$p = \overline{p}(t)$$

and applied to boundaries. The continuity equations at the boundary nodes are replaced by this condition.

Prescribed pressure condition is usually applied to confined flow problems to ensure a mathematically well-defined problem. Along open boundaries where the pressure is known, a boundary condition of a distributed normal-traction load is more appropriate.

ref. KJB Section 4.5

Similar to prescribed velocity conditions, the time birth and death options are available in prescribed pressure condition.

3.4.2.4. Zero pressure

Physically, this condition is equivalent to a prescribed zero pressure. However, a zero pressure condition removes the pressure degree of freedom and, therefore, the time birth and death options cannot be used.

This condition cannot be used in low-speed compressible flow models, since a zero pressure is not allowed.

3.4.2.5. Prescribed rotational velocity

In this condition, the velocity is prescribed by means of a time-dependent angular velocity $\bar{\boldsymbol{\Omega}}(t)$ (see also the figure below)

$$\mathbf{v}(t) = \bar{\boldsymbol{\Omega}}(t) \times (\mathbf{x} - \bar{\mathbf{x}}_0)$$

where, $\bar{\mathbf{x}}_0$ is the center of the rotation and $\mathbf{x}$ denotes the coordinates of the boundaries.

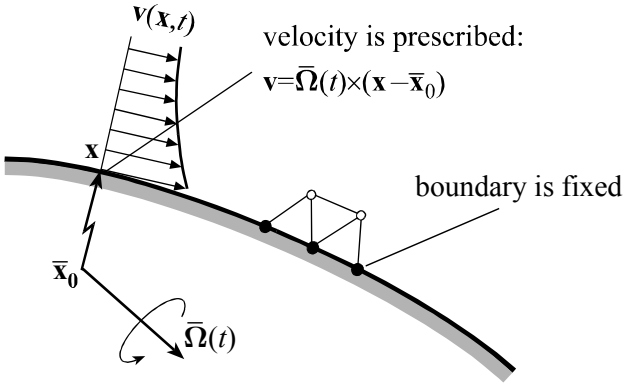


Figure 3.3 Prescribed rotational velocity

This condition can be applied to a wall where the solid boundary is rotating. It is important to understand that, since the Eulerian coordinate system is used for the fluid domain, the boundary nodes are fixed while the fluid velocity equals the solid velocity. Therefore, from a fluid point of view, the boundary is "fixed". Note that a moving boundary or a non-zero displacement condition cannot be modeled using this condition.

3.4.2.6 Concentrated force load

In this condition, a prescribed time-dependent force $\overline{\mathbf{F}}(t)$ is directly applied to boundaries. The force is added to the right-hand side of the momentum equations of the boundary nodes. Note that this condition can directly be applied to nodes as well. This force is also called external nodal force.

This condition has no effect on the nodes where a prescribed velocity condition is used since the momentum equation has been replaced by the velocity condition.

3.4.2.7 Distributed normal-traction load

This is one of the most important usual boundary conditions. It can only be applied to boundary lines and surfaces of two-dimensional and three-dimensional computational domains, respectively.

In this condition, a time-dependent normal stress (called normal traction) $\overline{\tau}_{nn}(t) = \mathbf{n} \bullet \boldsymbol{\tau} \bullet \mathbf{n}$ is prescribed. The stress is integrated to an equivalent nodal force condition

$$\mathbf{F}(t) = \int h^v \overline{\tau}_{nn}(t) d\mathbf{S}$$

and then added to the right-hand side of the momentum equations as the concentrated force load. Here h^v is the virtual quantity of velocity on the boundary.

Note that the normal stress consists of the pressure and the normal shear stress. Along open boundaries, the normal shear stress is usually negligible compared with the pressure. Therefore, a normal-traction is usually applied

to open boundaries where the pressure is known. In particular, when $\bar{\tau}_{nn}(t) = 0$, the application of the normal-traction loads is trivial because it is equivalent to no normal-traction loads.

This condition has no effect on the nodes where a normal velocity condition is prescribed, since the normal momentum equation has been replaced by the normal velocity condition.

3.4.2.8 Field centrifugal load

As described in Section 2.3, when the whole field is rotating, the flow equations can be formulated in a rotational reference coordinate system. The problem is then equivalent to a problem defined in a fixed coordinate system subjected to additional (centrifugal) forces. These forces can be written as

$$\mathbf{F}(t) = \int h^v \mathbf{f}_c \, dV$$

where h^v is the virtual quantity of velocity and $\mathbf{f}_c$ is defined in Chapter 2 but rewritten here, for the purpose of indicating the input parameters in the force definition

$$\mathbf{f}_c = -\rho \ddot{\bar{\mathbf{x}}}_0(t) - \rho \dot{\bar{\mathbf{\Omega}}}(t) \times (\mathbf{x} - \bar{\mathbf{x}}_0(t)) - 2\rho \bar{\mathbf{\Omega}}(t) \times \mathbf{v} - \rho \bar{\mathbf{\Omega}}(t) \times \bar{\mathbf{\Omega}}(t) \times (\mathbf{x} - \bar{\mathbf{x}}_0(t))$$

The center position $\bar{\mathbf{x}}_0(t)$ and the angular velocity $\bar{\mathbf{\Omega}}(t)$ are required to define this condition. In principle, the acceleration of the center position, $\ddot{\bar{\mathbf{x}}}_0(t)$, could be evaluated by the program. However, the numerical error could be severe for many applications. Therefore $\ddot{\bar{\mathbf{x}}}_0(t)$ is also required in the input.

Keep in mind that the velocity solved in this coordinate system is the relative velocity with respect to the rotational frame (which is denoted as $\mathbf{v}_r$ in Section 2.3).

Strictly speaking, this condition is not a “boundary” condition since the force is applied to the whole domain.

It is noteworthy to make a few remarks.

- Consider the case of zero angular velocity. The only term left in the load is the acceleration of the center position of the frame. This term can be used to model any field force load. For example, a time-dependent gravitational force can be modeled. In the meantime, the gravity must be specified as zero in the material data set to prevent a duplicated gravity load definition.
- In steady-state analyses, the first two terms vanish. No input of the center acceleration is required.

3.4.3 Special boundary conditions for fluid

3.4.3.1 Fixed wall

At the interface of a fluid and a fixed solid, no-slip or slip conditions are usually applied. On the fixed wall condition, the boundary is fixed. In other words, the boundary displacement is zero. Fixed wall conditions can only be applied to boundary lines and surfaces, respectively, of two-dimensional and three-dimensional computational domains.

No-slip condition on fixed walls

When a no-slip condition on fixed walls is applied, the fluid velocity vector on that wall is prescribed to be zero

$$\mathbf{v} = \mathbf{0}$$

This condition is usually applied to wall boundaries in viscous flows. It is clear that this condition is equivalent to applying a zero velocity or prescribed zero velocity to all components of the velocity.

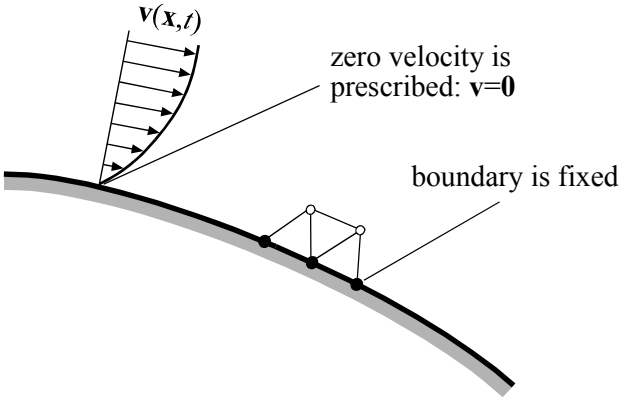


Figure 3.4 No-slip condition on fixed walls for incompressible, slightly compressible and low-speed compressible flows

Slip condition on fixed walls

On the other hand, when a slip wall condition is applied, the normal component of the velocity vector is prescribed to be zero

$$\mathbf{v} \cdot \mathbf{n} = 0$$

while the tangential components are free or computed as unknown variables from the governing equations.

In principle, this condition can be replaced by the condition of prescribed zero normal velocity. However, in the case of irregular boundaries, the procedure of defining the normal directions is tedious. It is much more convenient to use this slip condition.

This condition is usually applied to symmetric boundaries and to the wall boundaries where viscous effects are negligible. In certain applications, the computer capacity is limited such that large elements have to be used and, therefore, the boundary layers cannot be modeled. In this case, only because of the computer limitation, a slip condition may be used.

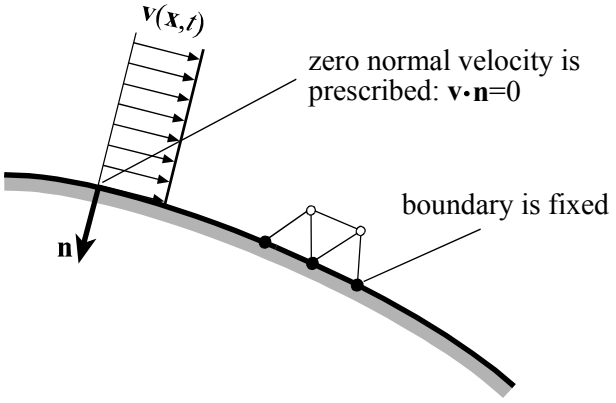


Figure 3.5 Slip condition on fixed walls for incompressible, slightly compressible and low-speed compressible flows

3.4.3.2 Uniform flow

The uniform flow condition is based on the assumption that the flow is uniform along the normal direction of a boundary, or

$$\frac{\partial \mathbf{v}}{\partial n} = \mathbf{0}$$

There are no parameters required in the input.

This condition can be applied to the boundaries at which the flow is almost uniform, for example, on a boundary representing "infinity". Uniform flow can only be applied to boundary lines and surfaces of two-dimensional and three-dimensional computational domains, respectively.

3.4.3.3 Angular velocity

This condition is similar to the prescribed rotational velocity, but can be applied to moving boundaries, where the coordinates of the boundary nodes vary with time (see figure below)

$$\mathbf{v}(t) = \bar{\mathbf{\Omega}}(t) \times \left(\mathbf{x}(t) - \bar{\mathbf{x}}_0 \right)$$

Here, we do not consider how the boundary nodes move. In a special case where the boundary is fixed, this condition is the same as a prescribed rotational velocity. Angular velocity can only be applied to boundary lines and surfaces of two-dimensional and three-dimensional computational domains, respectively.

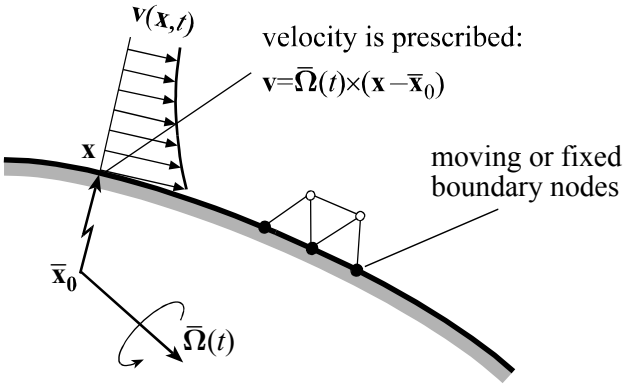


Figure 3.6 Angular velocity boundary condition for incompressible, slightly compressible and low-speed compressible flows

3.4.3.4 Moving wall

This condition broadens applications of the fixed wall condition to moving walls. Moving wall conditions can only be applied to boundary lines and surfaces of two-dimensional and three-dimensional computational domains, respectively.

No-slip condition on moving walls

The no-slip condition on a moving wall is the consistency between the fluid velocity and the solid velocity

$$\mathbf{v} - \dot{\bar{\mathbf{d}}}(t) = \mathbf{0}$$

This implies that the Lagrangian formulation is applied to all directions of the fluid boundary nodes. The boundary displacement $\bar{\mathbf{d}}(t)$ is required in the input of the condition, while the fluid velocity is computed based on the displacement.

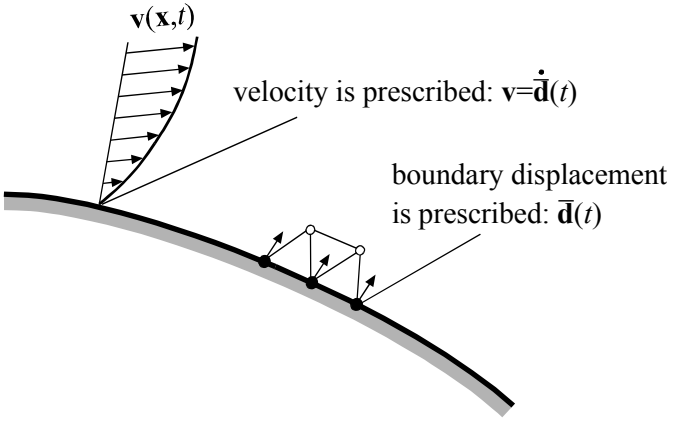


Figure 3.7 No-slip condition on moving walls for incompressible, slightly compressible and low-speed compressible flows

This condition is usually applied to moving wall boundaries in viscous flows. The dominant motion of the moving walls must be in the normal direction of the boundary. If the tangential movement is dominant, the moving wall condition with specified tangential velocity should be considered. They are introduced later in this section.

Slip condition on moving walls

In slip wall conditions, only the normal component of the velocity is enforced to be consistent with the solid counterpart

$$\left(\mathbf{v} - \dot{\bar{\mathbf{d}}}(t)\right)\bullet\mathbf{n} = \mathbf{0}$$

while the tangential components are free and computed as unknown variables from the governing equations. This implies that the Lagrangian formulation is used for only the normal direction of the fluid boundary nodes.

This condition is usually applied to the moving wall boundaries where viscous effects are negligible.

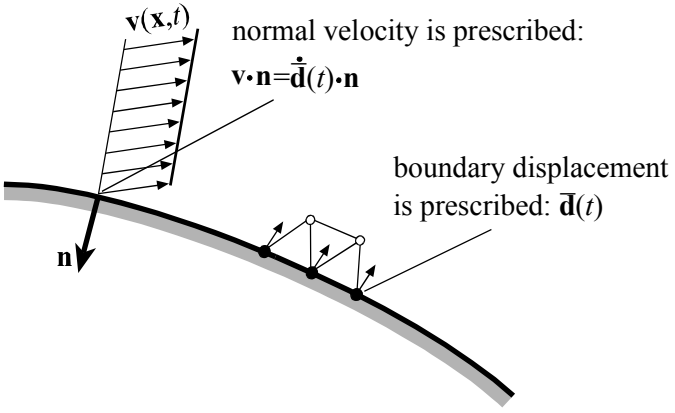


Figure 3.8 Slip condition on moving walls for incompressible, slightly compressible and low-speed compressible flows

In the no-slip moving wall boundary condition, the Lagrangian coordinate system is applied to all directions on the boundary. The condition sometimes presents difficulty in dealing with moving meshes if the tangential displacement is large. The use of a Lagrangian coordinate system in the tangential direction can result in large mesh distortions. However, the use of the Lagrangian formulation to that direction is not absolutely necessary. Most tangential movements do not change the shape of the computational fluid domain. Therefore, an ALE coordinate system can be applied to the moving wall. In particular, the displacement in the tangential direction can be completely avoided if the Eulerian formulation is applied to that direction.

In this ALE formulation, both the tangential velocity and the displacement vector are prescribed. They may or may not be consistent in the tangential direction. When they are consistent, the Lagrangian

formulation is implicitly used. When the tangential displacement is zero, the Eulerian formulation is then specified.

The new method is not only advantageous regarding the mesh quality, but also necessary for cases in which the validity of the mesh cannot be maintained.

Two such formulations are presented here.

No-slip condition on moving walls (type=tangential)

In this type of condition, the displacement is prescribed, denoted here as $\bar{\mathbf{d}}(t)$. This displacement causes the boundary to move. The normal fluid velocity is then computed based on the normal component of the displacement. The tangential velocity is prescribed separately (see figure below). This condition is

$$\mathbf{v} = \bar{v}(t)\boldsymbol{\tau} + \dot{\bar{\mathbf{d}}}(t)\bullet(\mathbf{I} - \boldsymbol{\tau}\boldsymbol{\tau})$$

where $\bar{v}(t)$ is the prescribed tangential velocity and $\boldsymbol{\tau}$ is the tangential direction that is computed by means of a specified direction $\bar{\mathbf{a}}$ and the current normal direction of the boundary $\mathbf{n}$

$$\boldsymbol{\tau} = \bar{\mathbf{a}} \times \mathbf{n}$$

Note that the tangential direction is not directly required, since it is, in general, not a constant vector.

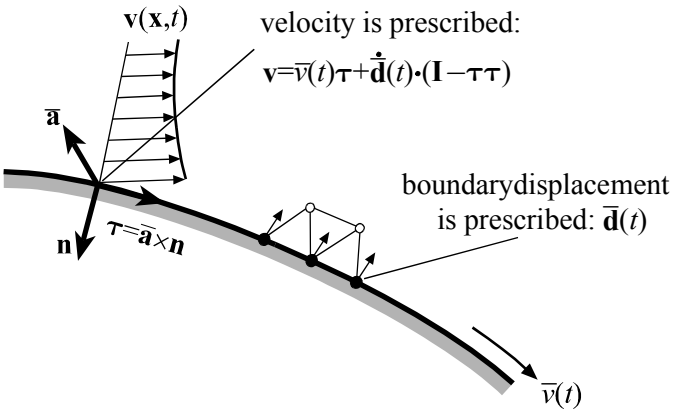


Figure 3.9 Moving wall condition (type=tangential) for incompressible, slightly compressible and low-speed compressible flows

No-slip condition on moving walls (type=rotational)

When the dominant boundary motion is determined by a rotation (see the next figure), we can apply a Eulerian formulation to that direction while leave the Lagrangian coordinate system in the other directions. In this case, the displacement and the angular velocity are prescribed. The velocity is computed as

$$\mathbf{v}(t) = \bar{\mathbf{\Omega}}(t) \times (\mathbf{x}(t) - \bar{\mathbf{x}}_0) + \dot{\bar{\mathbf{d}}}(t) \bullet (\mathbf{I} - \boldsymbol{\tau}\boldsymbol{\tau})$$

where $\bar{\mathbf{x}}_0$ is the center coordinates of the rotation, $\bar{\mathbf{\Omega}}$ is the angular velocity vector of the solid wall and the tangential direction $\boldsymbol{\tau}$ is computed as

$$\boldsymbol{\tau} = \frac{\bar{\mathbf{\Omega}}(t) \times (\mathbf{x}(t) - \bar{\mathbf{x}}_0)}{\left\| \bar{\mathbf{\Omega}}(t) \times (\mathbf{x}(t) - \bar{\mathbf{x}}_0) \right\|}$$

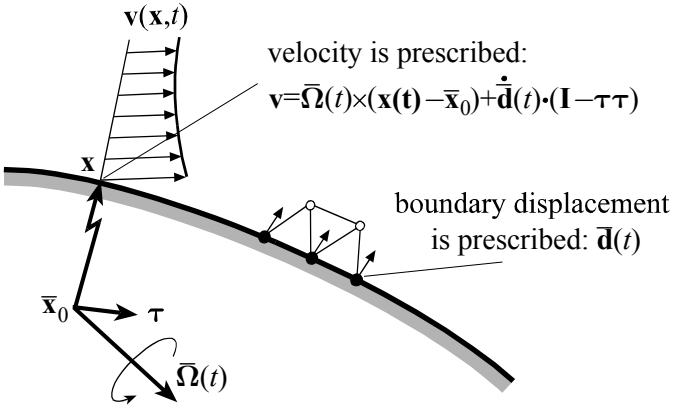


Figure 3.10 Moving wall condition (type=rotational) for

incompressible, slightly compressible and low-speed compressible flows

3.4.3.5 Fluid-structure interface

A fluid-structure interface is a moving wall for which the interface displacement is the solution of a solid model. However, fluid-structure-interaction means much more than just specifying an interface.

First, a solid model must have been created to which the fluid is coupled. In this solid model, fluid-structure interfaces must be specified corresponding to the interfaces specified in the fluid model, so that the program knows which parts of the fluid and solid models are interacting.

Second, since the Lagrangian formulation is used along the interface, the displacement as well as the fluid velocity are determined by the solid solution on the interface. This condition is called the kinematic condition of the fluid model. On the other hand, the fluid force must be applied to the solid interface to ensure the force balance on the interface. This condition is called the dynamic condition of the solid model.

Third, the nonlinear-coupled system must be solved to ensure that the kinematic and dynamic conditions are satisfied. The methods can be either iterative (between the fluid model and the solid model), or direct (a combined matrix system is solved).

In this section, we only present the definition of the fluid kinematic conditions and leave the details of the complete solution procedure of fluid-structure-interaction to Chapter 9.

The kinematic condition of the fluid-structure interface is exactly parallel to the moving wall boundary condition, except that the displacement $\bar{\mathbf{d}}(t)$ is part of the solution from the solid model, rather than being prescribed. The condition is also classified as a no-slip condition, a slip condition and no-slip conditions with prescribed tangential or rotational velocities. Of course, in the case where a tangential or a rotational velocity is prescribed, the displacement of the fluid nodes in that direction is generally (but not necessarily) zero.

This condition can only be applied to boundary lines and surfaces of two-dimensional and three-dimensional computational domains, respectively. The boundary geometries must coincide with their counterparts that are defined in the solid model.

3.4.3.6 Free surface

A free surface boundary is an interface between a liquid and a gas that has a negligible mass density and is thus treated as a vacuum. The free surface can be subjected to an ambient pressure and a surface tension. This condition can only be applied to boundary lines and surfaces, respectively, of two-dimensional and three-dimensional computational domain.

A free surface is a moving boundary. Two conditions are then applied: the kinematic and dynamic conditions.

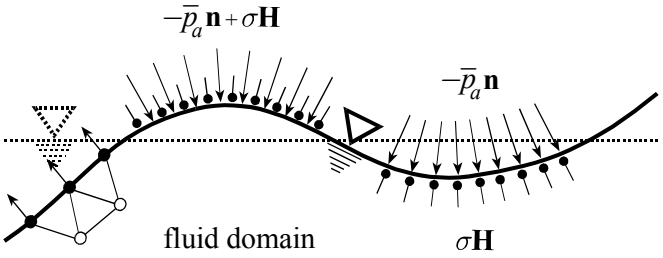


Figure 3.11 Sketch of a free surface condition

The kinematic condition of a free surface is that the fluid particle cannot move out of the surface or the normal velocity of the free surface must be the same as the normal velocity of the fluid

$$(\mathbf{v}-\dot{\mathbf{d}})\bullet\mathbf{n}=0$$

where $\mathbf{d}$ is the displacement of the free surface. However, this equation alone cannot define the displacement of the free surface. Assume that the tangential displacement is in the Eulerian coordinate system while we leave the normal displacement in the Lagrangian coordinate system and let it be denoted as S. In the originally fixed coordinate system (Eulerian), the kinematic condition becomes

$$\frac{\partial S}{\partial t}+\mathbf{v}\bullet\nabla S=0$$

This equation, in a moving coordinate system (here it is the free surface), is simply a coordinate transformation (see Section 2.14 for details),

$$\frac{\partial S}{\partial \tau} + (\mathbf{v} - \mathbf{w}) \bullet \nabla S = 0 \qquad (0.17)$$

We remark that the free surface position is steady here with respect to the new time frame τ and $\nabla S \sim \mathbf{n}$.

The kinematic condition states that the normal velocity of a fluid particle on the free surface is equal to the normal velocity of the free surface itself at the point where the particle is located.

The dynamic boundary condition on the free surface (applied to the momentum equations of the fluid) is the ambient pressure together with the surface tension

$$\boldsymbol{\tau}_n = -\overline{p}_a \mathbf{n} + \sigma \mathbf{H}$$

where $\overline{p}_a$ is the ambient pressure, σ is the surface tension and $\mathbf{H}$ is the curvature vector of the free surface. Let the free surface be $\mathbf{r} = \mathbf{r}(s_1, s_2)$, expressed in a local coordinate system $s_i, (i = 1, 2)$, we have

$$\mathbf{H} = \nabla^2 \mathbf{r} = \frac{1}{J} \nabla_i \left(J g^{ij} \nabla_j \mathbf{r} \right)$$

where

$$\left(g^{ij}\right) = \left(g_{ij}\right)^{-1} = \begin{pmatrix} \dfrac{\partial \mathbf{r}}{\partial s_1} \bullet \dfrac{\partial \mathbf{r}}{\partial s_1} & \dfrac{\partial \mathbf{r}}{\partial s_1} \bullet \dfrac{\partial \mathbf{r}}{\partial s_2} \\ \dfrac{\partial \mathbf{r}}{\partial s_1} \bullet \dfrac{\partial \mathbf{r}}{\partial s_2} & \dfrac{\partial \mathbf{r}}{\partial s_2} \bullet \dfrac{\partial \mathbf{r}}{\partial s_2} \end{pmatrix}^{-1}$$

$$J = \sqrt{g_{11} g_{22} - g_{12} g_{21}}$$

In particular, in a two dimensional space, $\mathbf{r} = \mathbf{r}(s_1)$,

$$\mathbf{H} = \frac{1}{r^m \|\mathbf{r}'\|}\left[\left(\frac{r^m \mathbf{r}'}{\|\mathbf{r}'\|}\right)' - m\|\mathbf{r}'\|\mathbf{e}_2\right]$$

where $m = 0$ and 1 in planar and axisymmetric cases, respectively.

It is important to note that, in order to obtain accurate solution of free surface problems, time step length must be sufficiently small (usually Courant number $v\Delta t/\Delta x \approx 0.1$).

3.4.3.7 Fluid-fluid interface

A fluid-fluid interface is defined between two fluids. It is a moving boundary condition. Two boundary conditions are then applied: the kinematic and dynamic conditions. This condition can only be applied to internal lines and surfaces, respectively, of two-dimensional and three-dimensional computational domain.

The kinematic boundary condition is the same as for a free surface.

The dynamic boundary condition on the interface is

$$\boldsymbol{\tau}_n^{(1)} - \boldsymbol{\tau}_n^{(2)} = \sigma\mathbf{H}$$

where σ is the surface tension, $\boldsymbol{\tau}_n^{(1)}$ and $\boldsymbol{\tau}_n^{(2)}$ are the normal components of the stress tensors in fluids 1 and 2 respectively, and $\mathbf{H}$ is the curvature vector of the interface. The dynamic condition indicates that a surface tension force is applied to the interface between two fluids. The pressure and velocity are assumed to be continuous across the interface.

3.4.3.8 Phase-change boundary

This condition is applied to problems involving changes of phase between liquid and solid, such as those occurring in crystal growths, metal castings, etc. This condition can only be applied to internal lines and surfaces of two and dimensional computational domains, respectively. These internal geometries must be shared with the (ADINA-F) solid and

fluid element groups. The solidification temperature must be prescribed as an additional usual boundary condition on the interface.

The phase-change boundary is also a moving boundary condition. The kinematic condition is that the liquid obeys a no-slip wall condition. The dynamic boundary condition on the interface is obtained by using the mass flux conservation and energy conservation (including latent heat release across the interface). The resultant condition becomes

$$\left(\frac{\mathbf{q}_f - \mathbf{q}_s}{\bar{L}\rho_s} - \dot{\mathbf{d}} \right) \bullet \mathbf{n} = 0$$

where $\mathbf{d}$ is the interface position vector, $\mathbf{q}_f$ and $\mathbf{q}_s$ are the heat fluxes in the fluid and the solid domains, respectively, ρ_s is the solid density, $\bar{L}$ is the latent heat per unit volume and $\mathbf{n}$ is the normal direction of the interface.

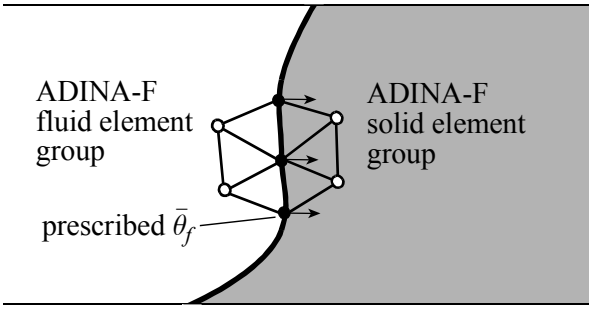


Figure 3.12 A typical phase-change boundary condition

3.4.3.9 Gap condition

The gap is an interface between two fluid domains. This condition can only be applied to internal lines and surfaces of two-dimensional and three-dimensional computational domains, respectively.

The gap's status may change due to specified physical conditions, indicating the connection or disconnection of the two domains. When they

are connected (the gap status is called open), the fluid can flow across the interface without involving any treatment of boundary conditions. In this case, the fluid variables are continuous across the interface. When the two domains are disconnected (the gap status is called closed), the fluid cannot flow across the interface. The gap condition functions as a no-slip wall condition to the fluid equations and as a zero heat flux condition to the temperature equation for both sides of the fluid domains. In this case, therefore, some solution variables are generally discontinuous (of course, the velocities are zero and continuous).

A typical application of the gap condition is shown in the next figure. The enlarged figures indicate the element connections when the gap is open and closed. When the gap is closed, the elements are disconnected, representing the disconnection of the two fluid domains. To achieve this condition, additional nodes are automatically generated at the same locations as the originally generated nodes (the figure shows here that they are shifted slightly purely for the purpose of clarification of disconnection). When the gap is open, the solution variables on these additional nodes are constrained to their original counterparts. Hence, continuous solution variables are modeled.

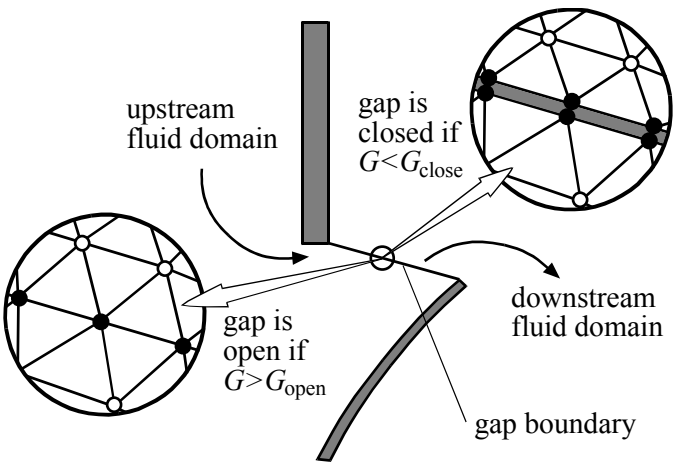


Figure 3.13 A typical gap boundary condition

The open/close status is controlled by certain criteria. Three physical conditions are currently available to control the gap status:

1) Gap status is controlled by the gap size G. If the gap is initially closed, it remains closed until the gap size exceeds a specified value $\overline{G}_{open}$. Conversely, if the gap is initially open, it remains open until the gap size is less than a specified value $\overline{G}_{close}$. In a two dimensional domain (including planar and axisymmetric cases), the size is the length of the interface, while in a three dimensional domain, the size is the area of the interface. Hence, the gap condition of this type must always be associated with some moving mesh boundary conditions.
2) Gap status is controlled by a time function G. If the gap is initially closed, it remains closed until the value of the time function value exceeds a specified value $\overline{G}_{open}$. Conversely, if the gap is initially open, it remains open until the value of the time function is less than a specified value $\overline{G}_{close}$.
3) Gap status is controlled by the averaged pressure difference $G = p_u - p_d$, where p_u and p_d are the pressures upstream and downstream of the gap, respectively. If the gap is initially closed, it remains closed until the pressure difference exceeds a specified open value $\overline{G}_{open}$. However, once it is open, the gap cannot be closed. Recall that the solution variables are continuous across the gap, the pressure "difference" remains zero for the rest of solution times.

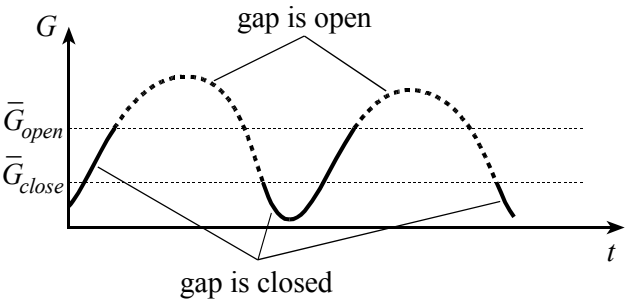


Figure 3.14 Diagram of gap open-close conditions

In order to prevent numerical oscillation in changing the gap status, the open value $\overline{G}_{open}$ must be slightly larger than the close value $\overline{G}_{close}$, usually a small percentage of $\overline{G}_{close}$.

3.4.3.10 Sliding-mesh interface

General description

Sliding-mesh capability is particularly useful for both fluid-flow and fluid-structure-interaction analyses that involve one or more moving parts. The sliding mesh boundary condition is designed to allow meshes in different regions to move relative to each other while the physical variables across the sliding interface remain continuous. There is a wide variety of fluid and FSI problems that can be solved using sliding mesh boundary conditions. Some typical applications include the flow in turbomachinery, mixing tanks and other devices that have one or more rotating parts as one example shown below:

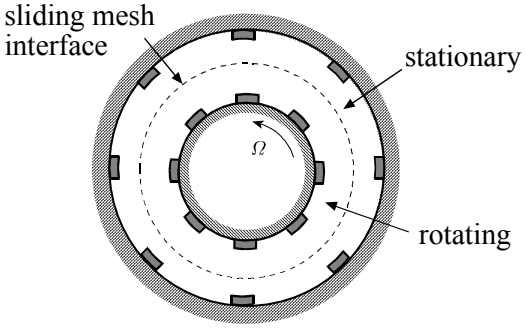


Figure 3.15 2D Rotor-Stator Interaction

Other problems that do not have rotating parts but translating moving parts may also require a sliding mesh boundary condition. One example, the two passing vehicles problem, is depicted in the following figure

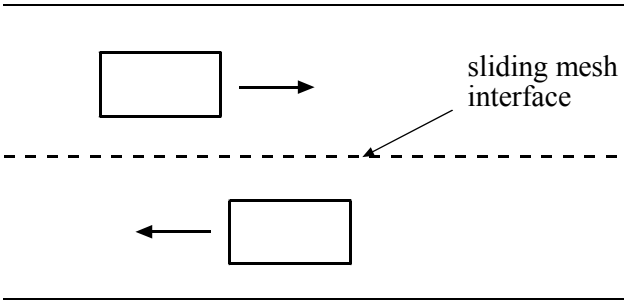


Figure 3.16 Two Passing Trains/Cars

Basic strategy and theory

It is assumed that in general the nodal points on both sides of the sliding interface are independent of each other. The numerical scheme at the interface should satisfy the conservation laws of mass, momentum and energy. For simplicity, the strategy used in a 2-D case depicted in the following figure is described. The nodal points M, S1 and S2 are sitting at different sides of the mesh interface. It is required to calculate the flux/force over the control volume represented by the node M. The control volume of M is composed of upper and lower sub-volumes. For the lower half, there is no difficulty to get the flux using the information from node M. However, for the upper half control volume, there is no direct information since the mesh is not connected across the interface. Therefore, the flux for the upper half control volume is calculated using the neighboring nodes S1 and S2.

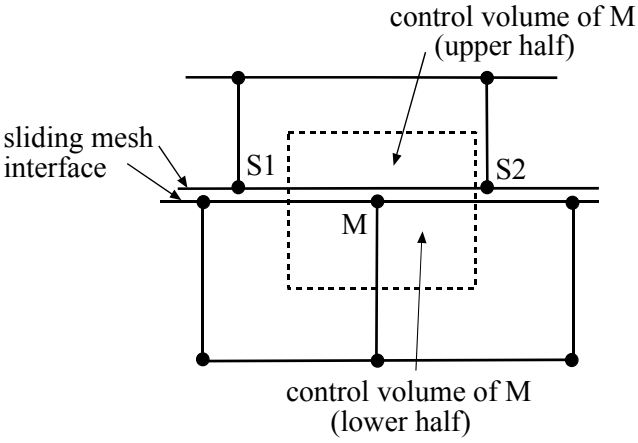


Figure 3.17 Sliding mesh interface coupling

Setting up a Sliding Mesh Model

The sliding mesh boundary condition needs to be applied on both sides of the zone interface. Two sets of geometry entities are required at the interface, having a common spatial location. The geometry entities at the interface can also have a small gap as shown in the figure

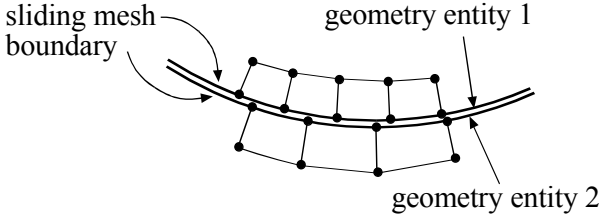


Figure 3.18 2-D geometry for sliding mesh interface

A sliding mesh boundary condition must be applied to both sides of the interface. The boundary condition applied to each side must be different, that is, they must have different numbers. After applying the sliding mesh boundary condition, a boundary condition pair has to be defined containing the sliding mesh boundary conditions previously defined. Since the sliding

mesh interface is only a mesh interface, physical variables are expected to be continuous across it. Therefore, the physical conditions at the edge of the sliding mesh interface must also be continuous. In other words, the physical condition at the edge of the two sides must be the same, as shown in the figure

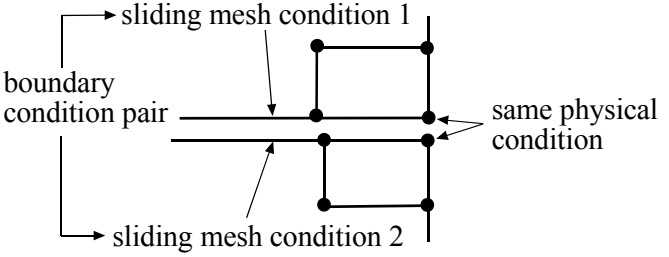


Figure 3.19 Physical conditions at edge of sliding mesh interface

In principle, the elements at two sides of the interface are independent, so different mesh subdivisions can be applied on each side. However, to obtain better convergence, it is recommended to use a similar element size at the interface. There are several ways to ensure that the nodal points are separate at the interface. For example, one can use different element groups for mesh generation and prevent same element group nodal coincidence checking at the interface.

Limitations for sliding mesh feature in current version

There are some situations where sliding mesh boundary conditions cannot be used. They are:

- It is available for FCBI and FCBI-C elements.
- For FCBI elements, it is only associated with the sparse solver
- It is not available for high-speed compressible flows.
- It cannot be used with the direct FSI coupling method.

3.4.3.11 Boundary friction condition

A general friction model has the form,

$$\mathbf{f} = -\tfrac{1}{2}\rho C_D \left(\mathbf{v}\bullet\mathbf{e}\right)^2 \mathbf{e}$$

where $\mathbf{f}$ is the nodal force per unit of area or length, and C_D is a drag coefficient and $\mathbf{e}$ is a direction vector. The nodal force is obtained by integration

$$\mathbf{F} = \int h^v \mathbf{f}\bullet d\mathbf{S}$$

where h^v is the virtual quantity of the velocity. The drag coefficient is assumed to be a function of the local Reynolds number, $C_D = a\left(\mathrm{Re}_L\right)^b$, where a, b user-supplied constants, $\mathrm{Re}_L = \rho\left|\mathbf{v}\bullet\mathbf{e}\right|L/\mu$, μ is the fluid viscosity, and L is a user-defined length scale. The direction of the friction force can be set to be the boundary normal, the velocity tangent or a user-supplied vector. Therefore, depending upon the choice of direction, the vector $\mathbf{e}$ becomes:

1. normal direction: $\mathbf{e} = \mathbf{n}$;
2. velocity tangential direction: $\mathbf{e} = \mathbf{v}\bullet\left(\mathbf{I}-\mathbf{nn}\right)\big/\left\|\mathbf{v}\bullet\left(\mathbf{I}-\mathbf{nn}\right)\right\|$;
3. user-defined vector: $\mathbf{e} =$ specified.

3.4.3.12 Vent boundary condition

Vents are regions where a flow resistance is imposed by the presence of a slot or other type of obstruction. The resistance to the flow is modeled by adding a force, $\mathbf{F}_{vent}$, to the right-hand side of the momentum equations,

$$\mathbf{f}_{vent} = -\tfrac{1}{2}\rho C_{loss}\left(\mathbf{v}\bullet\mathbf{n}\right)^2 \mathbf{n}$$
$$\mathbf{F}_{vent} = \int h^v \mathbf{f}_{vent}\bullet d\mathbf{S}$$

where ρ is the density of the fluid and C_{loss} is the loss coefficient associated with the vent, h^v is the virtual quantity of the velocity, and **n** is the surface normal.

The vent boundary condition is a special case of boundary friction, where the drag coefficient is a constant.

The user can also specify an arbitrary direction, **e**, instead of the boundary normal **n**. This is for cases where angled slots are present.

3.4.3.13 Fan boundary condition

The objective of the fan boundary condition is to provide a lumped-parameter model to predict the amount of flow through a fan, given its characteristic curve. A fan characteristic curve has the form,

$$p_s = f(Q) \geq 0$$

where $Q = \left| \int_{fan} \mathbf{v} \bullet d\mathbf{S} \right|$ is the flow rate (L^3 / t), p_s is the fan static pressure, $p_s = p_d - p_i$, p_d is the discharge pressure, and p_i is the inlet pressure.

The fan curve model in ADINA F is assumed to be a polynomial of the form,

$$f(Q) = C_0 + C_1 Q^{M_1} - C_2 Q^{M_2}$$

where C_0, C_1, C_2, M_1 and M_2 are user-specified constants.

The fan is modeled as an additional body force in the direction normal to the fan plane. Such force is proportional to the static pressure or the pressure rise due to the presence of the fan,

$$\mathbf{F}_{fan} = \int h^v f(Q) \mathbf{e} \, dS$$

The direction $\mathbf{e}$ depends on the type of fan,

(1) for intake fan: $\mathbf{e} = -\mathbf{n}$;
(2) for exhaust fan: $\mathbf{e} = \mathbf{n}$;
(3) for internal fan: $\mathbf{e} =$ specified

3.4.3.14 Multiple reference frames

Many fluid flow problems involve multiple rotating parts. For examples, rotor/stator interactions, multiple rotors (see figures below), etc., those widely exist in turbomachinery flows, mixing tanks and other related device. ADINA-F/FSI provides the Multiple Reference Frame (MRF) method for such problems. In the MRF approach, the interactions between the rotating parts are assumed weak and uniform, and therefore a steady-state interaction condition can be applied to the interface between different reference frames.

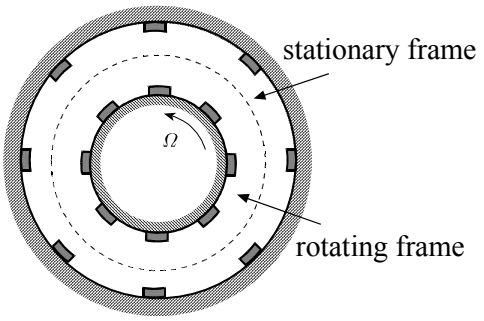


Figure 3.20 A weak rotor-stator interaction

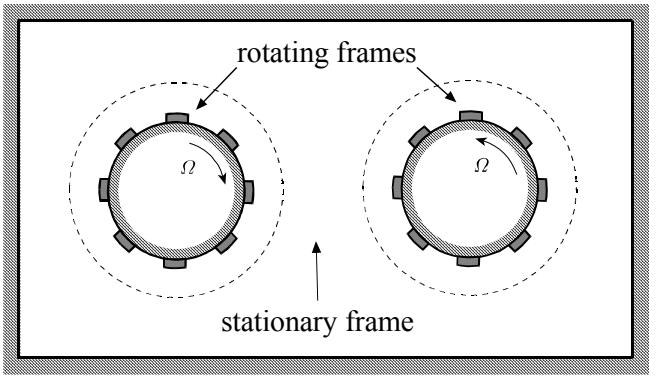


Figure 3.21 Two rotors in a chamber

With the MRF method, the computational domain is divided into a few sub-domains, each of them having different reference frames applied while the meshes are maintained stationary. Different frames are modeled using corresponding centrifugal forces as used in the single rotating model. The physical quantities on each interface are properly computed to ensure the conservations of mass, momentum, energy, etc.

In order to set up a MRF model, one has to first apply centrifugal loading (related to a specific reference frame) to a certain element group(s). Secondly, one has to apply a sliding-mesh condition on both interfaces that are connected (see the following figure). The sub-domains are completely separated and it is not required to have the interface mesh-nodes matched. One can also refer to the sliding-mesh condition for more explanations. Please note, the boundaries separating different reference frames should have zero normal frame velocity. For most cases, the boundary should be circular/cylindrical in the 2D/3D domain.

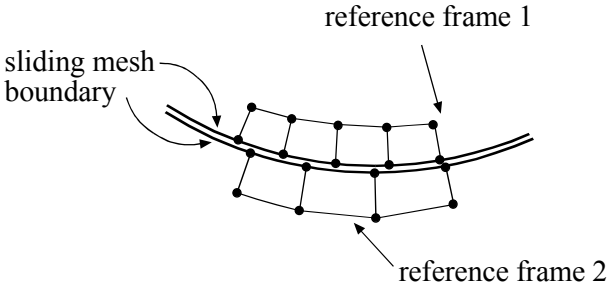


Figure 3.22 Model setup of a multiple-reference frame

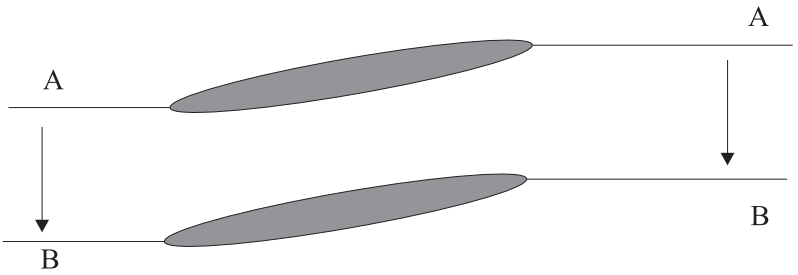
The MRF feature is currently only available for FCBI-C elements.

3.4.3.15 Periodic boundary condition

The periodic boundary condition is for cases where the physical geometry and the expecting flow pattern have a periodicity in space. Each condition is specified as a pair in which one boundary is transformed from the other. For example, as shown in the two figures in this section, the boundary of condition B is obtained by translating/rotating the boundary of condition A. Currently the types of transformation can be either translation or rotation. Meshes of the pair may be different.

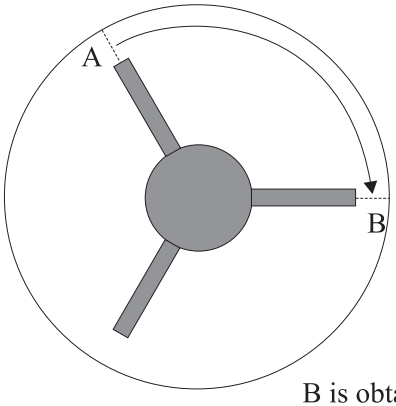
This is the procedure for creating a periodic boundary condition: (1) define periodic boundary condition A, in which the transformation from A to B is specified; (2) define periodic boundary condition B (in which the transformation is not required); (3) define a "BCD-PAIR" that is formed by A and B.

The periodic boundary condition is only available for FCBI-C elements.



B is obtained by translating A

Illustration of translating periodic boundary pair



B is obtained by rotating A

Illustration of rotating periodic boundary pair

3.4.4 Usual boundary conditions for temperature

3.4.4.1 Prescribed temperature

In this condition, a time-dependent temperature can be directly prescribed and applied to boundaries:

$$\theta = \overline{\theta}(t)$$

The energy equation at the boundary nodes is then replaced by this equation.

This condition is usually applied to boundaries where the temperature is known.

There is a time birth and death option associated with the prescribed temperature. If only the birth option is used, the prescribed condition is inactive until the solution time is larger than the birth time. If only the death option is used, the prescribed condition is active until the time is larger than the death time. If both the birth and death options are used, the prescribed condition is only active between the birth time and death time. Hence, the death time must be larger than the birth time. While the prescribed condition is inactive, the temperature degree of freedom is free.

3.4.4.2 Zero temperature

Physically, this condition is equivalent to a prescribed zero temperature. However, the zero temperature condition removes the temperature degree of freedom from the governing system of discretized equations. There are, therefore, no time birth and death options for this condition.

3.4.4.3 Concentrated heat flow load

In this condition, a prescribed time-dependent heat flow load $\bar{Q}(t)$ is directly applied to boundaries. The load is directly added to the right-hand side of the energy equations of boundary nodes. Note that heat flow loads can directly be applied to nodes as well.

This condition has no effect if applied to the boundary where a temperature condition is prescribed, since the energy equation has been replaced by the prescribed temperature condition.

3.4.4.4 Distributed heat flux load

This is one of the most important boundary conditions for the temperature solution. This condition can only be applied to boundary lines and surfaces of two-dimensional and three-dimensional computational domains, respectively.

This condition results in concentrated heat flow loads by specifying the normal component of the external heat flux $\overline{q}_n(t)$. The applied heat flux is then integrated to a heat load

$$Q(t) = \int h^\theta \overline{q}_n(t)\, dS$$

and added to the right-hand side of the energy equations as in the case of concentrated heat flow loads. Here h^θ is the virtual quantity of temperature on the boundary. Note that when $\overline{q}_n(t) = 0$, the application of the heat flux is trivial since it is equivalent to no heat flux loads.

This condition is usually applied to boundaries where the heat flux is known. The adiabatic boundary has a zero heat flux condition.

This condition has no effect if applied to the boundary where a temperature condition is prescribed since the energy equation has been replaced by the prescribed temperature condition.

3.4.5 Special boundary conditions for temperature

3.4.5.1 Convection

The heat flux from the external convection condition is defined as

$$q_n = \overline{h}\left(\overline{\theta}_e - \theta\right)$$

where $\overline{h}$ is a heat convection coefficient and $\overline{\theta}_e$ is an environmental temperature, both can be constant, time-dependent or temperature-dependent. The applied heat flow is then computed by

$$Q(t) = \int h^\theta q_n\, dS \qquad (0.18)$$

where h^θ is the virtual quantity of temperature on the boundary. Noting that the temperature θ is the unknown solution variable, it has therefore contributed terms to both the right-hand side and the coefficient matrix.

Convection condition is usually applied to a boundary that is exposed to an open environment, typically as shown in the following figure. This

condition can only be applied to boundary lines and surfaces of two-dimensional and three-dimensional computational domains, respectively.

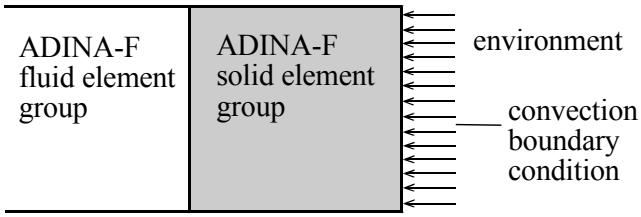


Figure 3.23 A typical convection boundary condition

3.4.5.2 Radiation

In this condition, the heat flux from an external radiation source is specified as

$$q_n = \overline{f}\,\overline{e}\left(R_e - R_t\right)$$

where R_e is the energy from the radiation source and R_t is the energy transmitted into the environment

$$R_e = \overline{\sigma}(\overline{c}_1\overline{\theta}_e + \overline{c}_2)^4, \quad R_t = \overline{\sigma}(\overline{c}_1\theta + \overline{c}_2)^4 \qquad (0.19)$$

in which, $\overline{\sigma}$ is the Stefan-Boltzmann constant, $\overline{f}$ is the shape factor, $\overline{e}$ is the emissivity, $\overline{\theta}_e$ is the temperature of the radiative source (or sink) , and $\overline{c}_1$ and $\overline{c}_2$ are two constants such that $\overline{c}_1\theta + \overline{c}_2$ has an absolute temperature scale. $\overline{e}$ and $\overline{\theta}_e$ can be constant, time-dependent or temperature-dependent. The conventional temperature unit is Kelvin in radiation, which corresponds to $\overline{c}_1 = 1$ and $\overline{c}_2 = 0$ if the temperature variable θ has the unit Kelvin. Some frequently used temperature units, when transformed to Kelvin or Rankine, are shown in the following table:

Table 3-2 Transformations of temperature units to absolute temperature units

$\theta \rightarrow$ $\overline{c}_1\theta + \overline{c}_2$ $\downarrow$	K	°R	°C	°F
K	$\overline{c}_1 = 1$ $\overline{c}_2 = 0$	$\overline{c}_1 = 5/9$ $\overline{c}_2 = 0$	$\overline{c}_1 = 1$ $\overline{c}_2 = 273.15$	$\overline{c}_1 = 5/9$ $\overline{c}_2 = 255.37$
°R	$\overline{c}_1 = 9/5$ $\overline{c}_2 = 0$	$\overline{c}_1 = 1$ $\overline{c}_2 = 0$	$\overline{c}_1 = 9/5$ $\overline{c}_2 = 491.67$	$\overline{c}_1 = 1$ $\overline{c}_2 = 459.67$

The heat flow is computed using Eq.(0.18) and discretized and assembled into the global matrix system.

The radiation condition is applied to the boundary that is exposed to a radiation source or sink. This condition can only be applied to boundary lines and surfaces of two-dimensional and three-dimensional computational domains, respectively.

3.4.5.3 Specular radiation

In some problems, certain solid boundaries have a high reflectivity in responding to radiative energies. Also, they may have certain percentages of energy emitted, diffused and transmitted. The specular radiation boundary condition is designed for these problems.

The following figure shows a portion of a surface surrounded by a fluid medium. The surface is assumed gray, diffused and specularly reflective, and may or may not be transparent. Consider an incident unit ray of energy, which is decomposed into four portions:

- a diffusely reflected portion $\overline{d}$;
- a transmitted portion $\overline{t}$;
- an absorbed portion which, by Kirchhoff's law, could be equal to the emittance *e*;
- a specularly reflected portion $\overline{s}$.

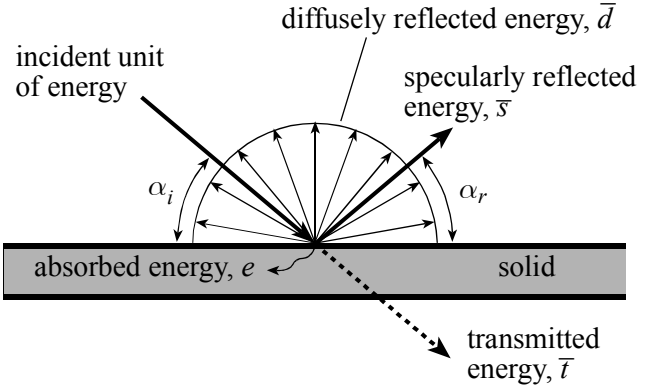


Figure 3.24 Schematic representation of radiation energies corresponding to an incident ray

The sum of the four portions satisfies

$$e + \overline{s} + \overline{d} + \overline{t} = 1 \tag{0.20}$$

The parameters, $\overline{s}$, $\overline{d}$ and $\overline{t}$, required in the input of this condition, can be constant, time-dependent or temperature-dependent. The parameter e is computed from Eq.(0.20). The transmitted energy could be lost to the environment and not traced in the solution, or could arrive at the next specular boundary and be reflected again. In addition, the specularly reflected energy is calculated in the solution algorithm by tracing the reflected ray of energy. The incident ray angle α_i and the reflected ray angle α_r are assumed equal.

Let R be the rate of outgoing radiant energy per unit area. It consists of the absorbed or emitted energy from the environment, transmitted energy and diffusely reflected energy,

$$R = eR_e + \overline{t}\,\overline{f}\,R_t + \overline{d}G \tag{0.21}$$

where R_e and R_t are defined in Eq.(0.19), G is the rate of all incoming radiant energies per unit area, $\overline{\sigma}$ is the Stefan-Boltzmann constant, $\overline{f}$ is

the shape factor, $\overline{\theta}_e$ is the temperature of the radiative source (or sink) , and $\overline{c}_1$ and $\overline{c}_2$ are two constants such that $\overline{c}_1\theta + \overline{c}_2$ has an absolute temperature scale (see previous section for details).

The rate of incoming radiant energy per unit area at any specular boundary element k is the sum of all rates of outgoing radiant energies from all specular radiation boundary elements

$$G_k = \sum_l \int_{S_l} F_{lk} R_l dS \tag{0.22}$$

where S_l is the area of the boundary element l and F_{lk} are the view factor matrices which are calculated using a ray tracing technique based on Lambert's law. It can be explained as the ratio of the energy that is emitted from element l and arrives at element k, directly and indirectly, to the energy emitted by point l. It is clear that the view factor matrix has a dimension $N_{es} \times N_{es}$, where N_{es} is the number of specular boundary elements.

Using Eq.(0.22) and applying the Galerkin method to Eq.(0.21), we obtain the finite element equation governing R

$$\int_S h^\theta \left(R - eR_e - \overline{t}\,\overline{f}\,R_t - \overline{d}G \right) dS = 0 \tag{0.23}$$

where h^θ is the virtual quantity of temperature on the boundary. This equation, called a radiosity equation, is linear and governs N_{ns} unknown variables R defined at specular boundary nodes.

In certain cases, not all specular boundary elements need to be coupled. For example, a domain is divided into two chambers that are separated by a solid wall as illustrated in the figure below. The specular boundaries in different chambers cannot see each other since no rays can penetrate through the solid wall (unless the ray is specified as being continuously transmitted from one side of the wall to the other side). This indicates that the view factor matrix has the form

$$\mathbf{F} = \begin{pmatrix} \mathbf{F}_1 & 0 \\ 0 & \mathbf{F}_2 \end{pmatrix}$$

with the two block diagonal matrices corresponding to the two chambers, respectively. The computation can then be carried out separately for the chambers. Therefore, we allow the specular boundary conditions to be divided into groups. Only specular boundary conditions within the same group specularly transmit energies to each other. Of course, energy is also transmitted by means of conduction and convection between the chambers.

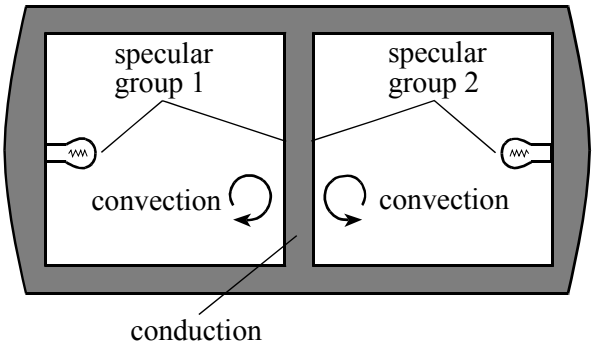


Figure 3.25 A typical usage of specular groups

With the variables R solved from Eq.(0.23) and G computed from Eq.(0.22), we can finally define the heat flux load that consists of absorbed incoming radiant energy and lost transmitted energy

$$q_n = e\left(G - R_t\right)$$

and the heat flow is then computed using Eq.(0.18).

It is clear then the application of the specular radiation boundary means much more than just specifying a regular boundary condition. Computing the heat flow load requires the solution of the variables R from a large linear system, which can be costly. Computational issues will be addressed in Section 13.5.

This condition can only be applied to lines and surfaces of two-dimensional and three-dimensional computational domains, respectively. These lines/surfaces must either be on the boundary or be interfaces between (ADINA-F) solid and fluid element groups.

3.4.6 Thermal resistance condition

In certain models where there is a very thin layer that has different heat conduction, this condition can be used to model the heat transfer through that layer. In this condition, a parameter $\overline{R}$ called thermal resistance must be specified. It is defined as

$$\overline{R} = \frac{\varepsilon}{k}$$

where k and ε are , respectively, the heat conductivity and thickness of the thin layer. A local analytical solution will then be applied to that layer.

Thermal resistance condition can be applied along lines (in 2D models) or along surfaces (in 3D models). These lines/surfaces can be located at boundaries or inside the computational domain.

3.4.7 User-supplied boundary condition

This is a specially defined user-supplied boundary condition. Similar to user-supplied material data, the boundary condition is programmed in Fortran by the user in a specially designed subroutine. This subroutine is invoked whenever the boundary condition is required.

In this subroutine, the solution variable identification number, the coordinates of the current spatial point, the element group, the element and the independent solutions, mass-ratios, etc., are provided as input arguments. In return, the subroutine will output the discretized boundary condition. This condition is represented in the form:

$$S = \overline{A} + \overline{\mathbf{B}} \bullet \mathbf{\Phi}$$

which is treated as a source of the corresponding equation from the boundary. Here, $\overline{A}$ is a scalar coefficient and $\overline{\mathbf{B}}$ is a coefficient vector and $\mathbf{\Phi}$ is the solution vector defined at the boundary node. The solution vector is defined as (based on the identification number I_{var} denoted as IVARID in the subroutine)

$$\mathbf{\Phi} = \begin{cases} u, v, w, p, \theta & \text{if } I_{\text{var}} = 1 \\ K, \varepsilon/\omega & \text{if } I_{\text{var}} = 2 \\ \phi_1, \phi_2, \ldots & \text{if } I_{\text{var}} = 3 \end{cases}$$

The values of $\overline{A}$ and $\overline{\mathbf{B}}$ will be assembled into the right-hand side of the discretized equation and the global matrix respectively as follows.

Corresponding to the current node, let i_k be the equation number of the current variable, i_1, i_2,... be the equation numbers of the solution variables $\mathbf{\Phi}$, and $A_{ij}\Delta X_j = B_i$ be the original equation system. The assemblage procedure in the Newton-Raphson method is

$$A_{i_k i_j} - \overline{B}_j \rightarrow A_{i_k i_j} \qquad j = 1,\ 2, \ldots$$
$$B_{i_k} + \overline{A} + \overline{B}_j \Phi_j \rightarrow B_{i_k}$$

As an example, if an external force, $\overline{F}(t) = a + bp$, is added to the y-momentum equation at boundary nodes where the user-supplied boundary condition l is applied, the coding would have the following structure in the Newton-Raphson iteration:

for boundary condition l
  for variable *identification* $I_{\text{var}} = 1$
    for variable $v\left\{\overline{A} = a + bp,\ \overline{B} = 0\right\}$

In cases where can be no confusion in the specified user-supplied boundary conditions, the condition checking can be omitted. For example, if only one condition is specified, in which only the force is specified, the above code can be simplified to

$$\overline{A} = a + bp, \quad \mathbf{B} = \mathbf{0}$$

It can be seen that advanced users can achieve much more with this capability. They can invoke their own programs to compute, possibly in a

complicated way, the value of the force and split it in any way they prefer. A different version of the program would be

- *compute a and b by any means possible*
- $\overline{A} = a + bp$
- $\overline{B}_i = \begin{cases} b & \text{if } i = k \\ 0 & \text{else} \end{cases}$

where, k is the index of the variable $\mathbf{\Phi}$ corresponding to the *y*-velocity (in this case, $k = 2$). This version would be more stable in general cases.

As another example, consider a pressure-dependent temperature is prescribed, $\overline{\theta}(p) = a + bp + cp^2$, at boundary nodes where the user-supplied boundary condition l is applied. Since $d\overline{\theta}/dp = b + 2cp$, the coding would have the following structure:

- *compute a, b and c by any means possible*
- $\overline{A} = 10^{21} \cdot (a + bp + cp^2)$
- $\overline{B}_i = 10^{21} \cdot \begin{cases} b + 2cp & \text{if } i = 4 \text{ (coefficient of } p) \\ -1 & \text{if } i = 5 \text{ (coefficient of } \theta) \\ 0 & \text{else} \end{cases}$

Where the large number 10^{21} is the pivot number added to ensure the specified condition is so dominant that the original terms in that equation become negligible.

The following is a sample example for 3 different user-supplied boundary conditions for mass ratios:

(1) Boundary condition label 1: prescribed $(\phi_1, \phi_2) = (0.2, 0.1)$
(2) Boundary condition label 2: prescribed $(\phi_1, \phi_2) = (0.1, 0.2)$
(3) Boundary condition label 4: apply the load that is defined as

$$q_i = -0.002 M_i \left(\prod_{j \neq i} \varphi_j + \varphi_i \right)$$

Here, the "mass-ratio" has been assumed to be the mass density ρ_i, so that

$$\varphi_i = \frac{C_i}{C} = \frac{\rho_i / M_i}{\sum_j \rho_j / M_j}$$

```
      SUBROUTINE ADFUBC(L,NL,R,NR,VAR,NVAR,A,B)
C=======================================================================
C
C
C     USER-SUPPLIED BOUNDARY CONDITIONS
C
C     THIS ROUTINE WILL BE CALLED AT EACH INTEGRATION POINT OF EVERY
C     BOUNDARY ELEMENT OF BOUNDARY CONDITION LABEL "IBCL" AT EACH
C     NONLINEAR EQUILIBRIUM ITERATION AT EACH TIME STEP.
C
C     HERE WE EXPRESS THE BOUNDARY CONDITION AS A SOURCE S, IMPOSED
C     INTO THE COMPUTATIONAL DOMAIN.
C
C     ------------------------------------------------------------------
C
C     PASSED IN:  L(NL),R(NR),VAR(NVAR)
C
C     PASSED OUT: A,B(NVAR)
C
C     ------------------------------------------------------------------
C     WHERE
C
C        L( 1) = IBCL   = BOUNDARY CONDITION LABEL (NAME IN THE
C                         BOUNDARY-CONDITION COMMAND)
C        L( 2) = IVARID = ID OF THE VARIABLE (=1,2 AND 3 FOR FLUID,
C                         TURBULENCE AND MASS (VARIABLE IN THE
C                         BOUNDARY-CONDITION COMMAND)
C        L( 3) = ICOMID = ID OF THE CURRENT EQUATION:
C                       = (1,2,3,4,5) FOR (U,V,W,P,T  ), IF IVARID=1
C                       = (1,2      ) FOR (K,E/W)      , IF IVARID=2
C                       = (1,2,...  ) FOR (S_1,S_2,...), IF IVARID=3
C        L( 4) = METHOD = ID OF THE ITERATION METHOD:
C                       = 0, IF SUCCESIVE SUBSTITUTION METHOD
C                       = 1, IF NEWTON METHOD
C        L( 5) = IELG   = ELEMENT GROUP NUMBER
C        L( 6) = IELM   = ELEMENT NUMBER
C        L( 7) = IU     = ENTRY OF THE X-VELOCITY       IN VAR(*)
C        L( 8) = IV     = ENTRY OF THE Y-VELOCITY       IN VAR(*)
C        L( 9) = IW     = ENTRY OF THE Z-VELOCITY       IN VAR(*)
C        L(10) = IP     = ENTRY OF THE PRESSURE         IN VAR(*)
C        L(11) = IT     = ENTRY OF THE TEMPERATURE      IN VAR(*)
C        L(12) = IK     = ENTRY OF THE TURBULENCE-K     IN VAR(*)
C        L(13) = IE     = ENTRY OF THE TURBULENCE-E/W   IN VAR(*)
C        L(14) = IX     = ENTRY OF THE X-COORDINATE     IN VAR(*)
C        L(15) = IY     = ENTRY OF THE Y-COORDINATE     IN VAR(*)
```

```
C       L(16) = IZ     = ENTRY OF THE Z-COORDINATE     IN VAR(*)
C       L(17) = IM     = ENTRY OF THE MASS SPECIES     IN VAR(*)
C       L(18) = IQ     = ENTRY OF THE CURRENT VARIABLE IN VAR(*)
C                         NOTE THAT IQ DEPENDS ON IVARID AND ICOMID:
C                       IQ = IU          IF(IVARID=1 AND ICOMID=1)
C                          = IV          IF(IVARID=1 AND ICOMID=2)
C                          = IW          IF(IVARID=1 AND ICOMID=3)
C                          = IP          IF(IVARID=1 AND ICOMID=4)
C                          = IT          IF(IVARID=1 AND ICOMID=5)
C                          = IK          IF(IVARID=2 AND ICOMID=1)
C                          = IE          IF(IVARID=2 AND ICOMID=2)
C                          = IM+ICOMID, IF(IVARID=3)
C       L(20) = JX     = ENTRY OF THE X-OUT-NORMAL     IN   R(*)
C       L(21) = JY     = ENTRY OF THE Y-OUT-NORMAL     IN   R(*)
C       L(22) = JZ     = ENTRY OF THE Z-OUT-NORMAL     IN   R(*)
C
C       VAR(IU)   = U  = X-VELOCITY
C       VAR(IV)   = V  = Y-VELOCITY
C       VAR(IW)   = W  = Z-VELOCITY
C       VAR(IP)   = P  = PRESSURE
C       VAR(IT)   = T  = TEMPERATURE
C       VAR(IK)   = TK = TURBULENCE_K
C       VAR(IE)   = TE = TURBULENCE_E OR TURBULENCE-W
C       VAR(IX)   = X  = X-COORDINATE
C       VAR(IY)   = Y  = Y-COORDINATE
C       VAR(IZ)   = Z  = Z-COORDINATE
C       VAR(IM)   = S_1= MASS-RATIO_0 (FLUID ITSELF)
C       VAR(IM+1) = S_2= MASS-RATIO_1
C       ......
C       VAR(IQ)   = VALUE OF THE CURRENT VARIABLE
C
C       R(JX)     = SX = X-OUT-NORMAL OF THE BOUNDARY
C       R(JY)     = SY = Y-OUT-NORMAL OF THE BOUNDARY
C       R(JZ)     = SZ = Z-OUT-NORMAL OF THE BOUNDARY
C
C     ---------------------------------------------------------------
C     THE DEFINITION OF THE PASSED OUT VARIABLES A AND B(*) DEPENDS ON
C     THE FLAGS "IVARID", "METHOD" AND THE FLOW TYPE (INCOMPRESSIBLE
C     OR COMPRESSIBLE).
C
C       +-----------------------------------------------------+
C       |                                                     |
C       |   CASE: FLUID OR TURBULENCE            (IVARID<=2)  |
C       |         SUCCESSIVE SUBSTITUTION METHOD (METHOD=0 )  |
C       |         INCOMPRESSIBLE FLOWS                        |
C       |                                                     |
C       +-----------------------------------------------------+
C
C        THE SOURCE S IS SPLIT INTO THE FORM
C
C           S  = A + B(IU)*U  + B(IV)*V  + B(IW)*W
C                  + B(IP)*P  + B(IT)*T
C                  + B(IK)*TK + B(IE)*TE
C
C        CAUTION: THERE ARE MANY WAYS TO SPLIT S. SOME OF
C                 THEM MAY CAUSE NUMERICAL INSTABILITY. IT
C                 IS RECOMMENDED TO ALWAYS KEEP B(IQ)<=0.
C
C       +-----------------------------------------------------+
C       |                                                     |
C       |   CASE: FLUID OR TURBULENCE            (IVARID<=2)  |
C       |         NEWTON METHOD                  (METHOD=1 )  |
C       |         INCOMPRESSIBLE FLOWS                        |
```

```
C       |                                                     |
C       +-----------------------------------------------------+
C
C        A     = THE SOURCE S
C        B(IU) = DERIVATIVE OF S WITH RESPECT TO U
C        B(IV) = DERIVATIVE OF S WITH RESPECT TO V
C        B(IW) = DERIVATIVE OF S WITH RESPECT TO W
C        B(IP) = DERIVATIVE OF S WITH RESPECT TO P
C        B(IT) = DERIVATIVE OF S WITH RESPECT TO T
C        B(IK) = DERIVATIVE OF S WITH RESPECT TO TK
C        B(IE) = DERIVATIVE OF S WITH RESPECT TO TE
C
C       +-----------------------------------------------------+
C       |                                                     |
C       |   CASE: FLUID OR TURBULENCE            (IVARID<=2)  |
C       |         SUCCESSIVE SUBSTITUTION METHOD (METHOD=0 )  |
C       |         COMPRESSIBLE FLOWS                          |
C       |                                                     |
C       +-----------------------------------------------------+
C
C        THE SOURCE S IS SPLIT INTO THE FORM
C
C           S  = A + B(IU)*DU + B(IV)*DV + B(IW)*DW
C                  + B(IP)*D  + B(IT)*E
C                  + B(IK)*TK + B(IE)*TE
C
C        WHERE, D=DENSITY, E=TOTAL ENERGY, DU=D*U, DV=D*V
C               AND DW=D*W.
C
C        CAUTION: THERE ARE MANY WAYS TO SPLIT S. SOME OF
C                 THEM MAY CAUSE NUMERICAL INSTABILITY. IT
C                 IS RECOMMENDED TO ALWAYS KEEP B(IQ)<=0.
C
C       +-----------------------------------------------------+
C       |                                                     |
C       |   CASE: FLUID OR TURBULENCE            (IVARID<=2)  |
C       |         NEWTON METHOD                  (METHOD=1 )  |
C       |         COMPRESSIBLE FLOWS                          |
C       |                                                     |
C       +-----------------------------------------------------+
C
C        A     = THE SOURCE S
C        B(IU) = DERIVATIVE OF S WITH RESPECT TO DU
C        B(IV) = DERIVATIVE OF S WITH RESPECT TO DV
C        B(IW) = DERIVATIVE OF S WITH RESPECT TO DW
C        B(IP) = DERIVATIVE OF S WITH RESPECT TO D
C        B(IT) = DERIVATIVE OF S WITH RESPECT TO E
C        B(IK) = DERIVATIVE OF S WITH RESPECT TO TK
C        B(IE) = DERIVATIVE OF S WITH RESPECT TO TE
C
C        WHERE, D=DENSITY, E=TOTAL ENERGY, DU=D*U, DV=D*V
C               AND DW=D*W.
C
C       +-----------------------------------------------------+
C       |                                                     |
C       |   CASE: MASS TRANSFERS                 (IVARID=3)   |
C       |         SUCCESSIVE SUBSTITUTION AND NEWTON METHODS  |
C       |         INCOMPRESSIBLE AND COMPRESSIBLE FLOWS       |
C       |                                                     |
C       +-----------------------------------------------------+
C
C        THE SOURCE S IS SPLIT INTO THE FORM
C
```

```
C           S  = A + B(IQ)*VAR(IQ)
C
C        CAUTION: THERE ARE MANY WAYS TO SPLIT S. SOME OF
C                 THEM MAY CAUSE NUMERICAL INSTABILITY. IT
C                 IS RECOMMENDED TO ALWAYS KEEP B(IQ)<=0.
C
C=======================================================================
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION  L(NL),R(NR),VAR(NVAR),B(NVAR)
C-----------------------------------------------------------------------
C     START TO CODE HERE
C-----------------------------------------------------------------------
      PARAMETER (NMASS=2)
      DIMENSION  XM(NMASS),X(NMASS)
C-----------------------------------------------------------------------
      R(1)   =   R(1)
C-----------------------------------------------------------------------
      IBCL   =   L( 1)
      IVARID =   L( 2)
      ICOMID =   L( 3)
C     METHOD =   L( 4)
      IM     =   L(17)
      IQ     =   L(18)
      UNDER  =   1.D-20
C
C=======================================================================
C
      IF(IVARID.EQ.1) THEN
C
C     +-----------------------------------------------------------+
C     |                                                           |
C     |  FLUID BOUNDARY CONDITIONS ARE CODED HERE (U,V,W,P,T)     |
C     |                                                           |
C     +-----------------------------------------------------------+
C
C
C        +-----------------------------------------------------+
C        |  END OF FLUID BOUNDARY CONDITIONS HERE              |
C        +-----------------------------------------------------+
C
         GO TO 9999
      ENDIF
C
C=======================================================================
C
      IF(IVARID.EQ.2) THEN
C
C     +-----------------------------------------------------------+
C     |                                                           |
C     |  TURBULENCE BOUNDARY CONDITIONS ARE CODED HERE (K,E/W)    |
C     |                                                           |
C     +-----------------------------------------------------------+
C
C
C        +-----------------------------------------------------+
C        |  END OF RTURBULENCE BOUNDARY CONDITIONS HERE        |
C        +-----------------------------------------------------+
C
         GO TO 9999
      ENDIF
C
C=======================================================================
C
```

```
      IF(IVARID.EQ.3) THEN
C
C     +-------------------------------------------------------------+
C     |                                                             |
C     |  MASS TRANSFER BOUNDARY CONDITIONS ARE CODED HERE           |
C     |                                                             |
C     |  THE FOLLOWING EXAMPLE IS FOR 3 BOUNDARY CONDITIONS:        |
C     |                                                             |
C     |  IBCL = 1:  SPECIFY THE MASS RATIO 0.2                      |
C     |  IBCL = 2:  SPECIFY THE MASS RATIO 0.1                      |
C     |  IBCL = 4:  CVD CONDITION                                   |
C     |                                                             |
C     |  NMASS  =   NUMBER OF REACTING SPECIES (=2 HERE)            |
C     |  XM(*)  =   MOLECULAR WEIGHT                                |
C     |  X(*)   =   MOLE FRACTION                                   |
C     |  XNU    =   STOICHIOMETRIC COEFFICIENT                      |
C     |  XKA    =   KENETIC REACTION RATE CONSTANT                  |
C     |  NOTE: THE LARGE NUMBER 1.D+20 USED HERE IS TO FORCE THE    |
C     |        SPECIFIED VALUE TO BE RESPECTED IN THE FINAL EQUATION|
C     |                                                             |
C     +-------------------------------------------------------------+
C
         IF(IBCL.EQ.1) THEN
            IF(ICOMID.EQ.1) THEN
               SS =   0.2D0
            ELSE
               SS =   0.1D0
            ENDIF
            B(IQ) = - 1.D+20
            A     =   1.D+20*SS
            GO TO 9999
         ENDIF
C
         IF(IBCL.EQ.2) THEN
            IF(ICOMID.EQ.1) THEN
               SS =   0.1D0
            ELSE
               SS =   0.2D0
            ENDIF
            B(IQ) = - 1.D+20
            A     =   1.D+20*SS
            GO TO 9999
         ENDIF
C
         IF(IBCL.EQ.4) THEN
            XM(1) =   2.D0
            XM(2) =   1.D0
            XNU   =   1.0D0
            XKA   =   0.002D0
C
C           COMPUTE X(*)
C
            SUM      =   0.D0
            DO 1 I   =   1, NMASS
            X(I)     =   VAR(IM+I)/XM(I)
            SUM      =   SUM + X(I)
 1          CONTINUE
            SUM      =   MAX(SUM,UNDER)
            DO 2 I   =   1, NMASS
            X(I)     =   X(I)/SUM
 2          CONTINUE
C
C           COMPUTE A AND B
```

```
C
            XX       =   1.D0
            DO 3 I   =   1, NMASS
            IF(I.EQ.ICOMID) GO TO 3
            XX       =   XX*X(I)
 3          CONTINUE
            FAC      =   XNU*XKA
            B(IQ)    = - FAC/SUM
            A        = - FAC*XM(ICOMID)*XX
            GO TO 9999
          ENDIF
C
C         +------------------------------------------------------------+
C         |  END OF MASS BOUNDARY CONDITIONS HERE                      |
C         +------------------------------------------------------------+
C
      ENDIF
C-----------------------------------------------------------------------
 9999 CONTINUE
      RETURN
      END
```

3.4.8 Shell-thermal condition

In certain models where a thin shell structure is present, this condition can be used to model the heat transfer through that layer. The shell deformation may be neglected or computed (as an TFSI problem). This condition is classified into three types, namely boundary-convection, heat-flux and internal-convection. It can be applied to model shell structure inside fluid domain or on fluid boundary.

The boundary-convection condition is defined as

$$q_b = \overline{k}_s / \overline{d}_s \left(\overline{\theta}_e - \theta \right)$$

where, $\overline{k}_s$ is shell thermal conduction, $\overline{d}_s$ is shell thickness and $\overline{\theta}_e$ is the environmental temperature. When applied to a boundary of a pure fluid problem, this condition is equivalent to a convection condition (observing that $\overline{h}_s = \overline{k}_s / \overline{d}_s$ is convection coefficient). When this condition is applied to a fluid-shell interface in TFSI problems, the thermal results are automatically used in determining the shell deformation.

The heat-flux condition is the same as the boundary-convection condition, except that the heat-flux $q_b = \overline{q}_b$ is directly specified.

The internal-convection condition is to link heat transfers between two fluid domains separated by a thin shell. For convenience of explanation, we will refer to the two domains as left and right respectively. The condition that is applied to the left fluid boundary is defined as

$$q_i = \overline{k}_s / \overline{d}_s \left(\theta_R - \theta_L \right)$$

where, θ_L and θ_R are fluid boundary temperatures on the left and right of the shell respectively. Similarly, $-q_i$ is applied to the fluid boundary on the right of the shell. We note that the left and right fluid boundaries must be separated (different nodes, elements, etc.), but share the same geometric location (where the shell is). The shell deformation may or may not be modeled. If the shell is modeled, the thermal results are automatically used in determining the shell deformation. For more information on modeling TFSI problems, refer to the fluid-structure interaction chapter.

3.4.9 Mass-flow-rate condition

Mass flow rate is defined as

$$\dot{m} = \left| \int_S \rho \mathbf{v} \cdot d\mathbf{S} \right|$$

where ρ is density, $\mathbf{v}$ is fluid velocity and $\mathbf{S}$ is the surface vector. The volume flow rate is defined as

$$\dot{V} = \left| \int_S \mathbf{v} \cdot d\mathbf{S} \right|$$

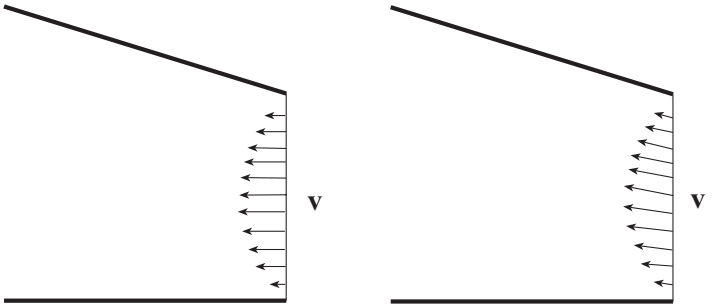
This condition provides a reasonable velocity profile such that the rate equals the specified rate. The flow direction can be specified in three methods below:

(1) A given vector: the direction is given by this vector.

(2) Inflow-normal: the direction is normal to the boundary and flow into the fluid domain.

(3) Outflow-normal: the direction is normal to the boundary and flow out of the fluid domain.

User can further request to adjust the specified direction according to the adjacent boundary direction of the current boundary, shown in the following figure.



velocity profile without boundary adjustment

velocity profile with boundary adjustment

This condition is currently only applicable to FCBI-C elements. For direction adjustment, the surrounding boundaries must be wall or FSI conditions.

3.5 Initial conditions

In transient analyses, all solution variables must be specified. The default initial conditions to all variables are zero values. Although initial conditions are not required in steady-state analyses, they are used as a "guessed" solution at the start of the equilibrium iterations. A good initial condition, as will be seen in Chapter 13, may not only accelerate the convergence during equilibrium iterations, but also be a key factor in obtaining solutions in certain cases.

In low-speed compressible flows, recalling that pressure, temperature and density must satisfy the state equation of the ideal gas law, zero pressure or zero temperature must not be specified. Since the default initial conditions to all variables are zeros, initial conditions are always required in both transient and steady-state analyses in compressible flow models.

Initial conditions can be improved in restart analyses or using the solutions mapped from a similar solution.

3.6 Material models

3.6.1 Constant material model

This is the simplest yet most frequently used material model. In this model, all fluid properties are assumed to be constant. They are

ρ = fluid density
μ = fluid viscosity
$\mathbf{g}$ = gravitational acceleration vector
C_p = specific heat at constant pressure
C_v = specific heat at constant volume
k = thermal conductivity coefficient
q^B = rate of heat generated per unit volume
β = thermal expansion coefficient
θ_0 = reference temperature in buoyancy force
σ = coefficient of surface tension
κ = bulk modulus of elasticity

The default values of these parameters are all zeros except the κ that is 10^{20}. Note that not all these parameters are required for a given problem. The parameters that must be input depend on the problem to be solved.

In incompressible flows, the bulk modulus of elasticity κ and the specific heat at constant volume C_v are not required. In this case, κ is

assumed to be infinite and C_v is forced to be equal to C_p. If heat transfer is not considered, C_p, k, q^B, β, and θ_0 can be further ignored.

In slightly compressible flows, C_v is forced to be equal to C_p and, therefore, can be ignored in the input. C_p, k, q^B, β and θ_0 can also be omitted if heat transfer is not included.

In low-speed compressible flows, κ is not used and therefore is ignored. The density is determined through the state equation as a function of pressure and temperature and thus not required. All other material data must be input.

The above notation will be used throughout the chapter in other material models.

3.6.2 Time-dependent model

This model contains the same parameters as those in the constant model. However, the following parameters in this model depend on time

$$\mu,\ C_p,\ C_v,\ k,\ \beta,\ q^B,\ \sigma,\ \kappa = f(t)$$

Besides its application in truly time-dependent materials, this type of material is also very useful in dealing with complex flows with constant properties. For example, with a time-dependent viscosity of decreasing-value in a steady-state analysis, the solution obtained at an earlier time step is the solution of a lower Reynolds number. It provides automatically a good initial condition for the next time step, which is a problem of a higher Reynolds number.

3.6.3 Power-law model

The viscosity of the power-law model is defined as

$$\mu = \begin{cases} \max\left\{AD^n, \mu_0\right\} & \text{if } n \geq 0 \\ \min\left\{AD^n, \mu_0\right\} & \text{if } n < 0 \end{cases}$$

where A, μ_0 and n are constants and D is the effective deformation rate. All other parameters are the same as those in the constant model.

The power-law model is a typical non-Newtonian fluid model. The definitions of A and a can be determined from the fundamental constitutive relation, in terms of shear stress and deformation rate,

$$\sigma = \mu D = AD^{n+1}$$

The parameter μ_0 serves as a lower bound (if n is non-negative) or an upper bound (if n is negative) of the viscosity.

3.6.4 Carreau model

This model is an extension of the power-law model. It sets both upper and lower bounds of the viscosity as defined in

$$\mu = \mu_\infty + \left(\mu_0 - \mu_\infty\right)\left(1 + AD^2\right)^n$$

where A, μ_0 and n are constants and D is the effective deformation rate. All other parameters are the same as those in the constant model.

Similar to the power-law model, the Carreau model is also a non-Newtonian fluid model.

3.6.5 Temperature-dependent model

This model contains the same parameters as those in the constant material condition model. However, the following parameters in this model depend on temperature

$$\mu,\ C_p,\ C_v,\ k,\ \beta,\ q^B,\ \sigma,\ \kappa = f(\theta)$$

This material model can only be applied to problems with heat transfer.

3.6.6 Temperature-dependent power-law model

This model is another extension of the power-law model. It includes the thermal effect in the viscosity as defined in

$$\mu = \begin{cases} \max\left\{ AD^n e^{C\theta}, \mu_0 \right\} & \text{if } n \geq 0 \\ \min\left\{ AD^n e^{C\theta}, \mu_0 \right\} & \text{if } n < 0 \end{cases}$$

where A, μ_0 and C are constants and D is the effective deformation rate. All other parameters are the same as those in the constant model.

Similar to the power-law model, this model is also a non-Newtonian fluid model. However, it can only be applied to problems with heat transfer.

3.6.7 Second order model

This model is an extension of the temperature-dependent power-law model. In this model, the viscosity is defined as

$$\mu = \begin{cases} \max\left\{ AD^n e^{C\theta + E\theta^2}, \mu_0 \right\} & \text{if } n \geq 0 \\ \min\left\{ AD^n e^{C\theta + E\theta^2}, \mu_0 \right\} & \text{if } n < 0 \end{cases}$$

with

$$n = F + G \ln D + H\theta$$

where A, μ_0, C, E, F, G and H are constants and D is the effective deformation rate. All other parameters are the same as those in the constant model.

Similar to the temperature-dependent power-law model, this model is also a non-Newtonian fluid model and can only be applied to problems with heat transfer.

3.6.8 Large-eddy-simulation model

The large-eddy-simulation (LES) model is a zeroth order turbulence model. Since it has no additional solution variables as introduced in the K-ε model, it is still treated as a "laminar" material model. The viscosity and the heat conductivity of the LES model are defined as

$$\mu = \mu_0 + \mu_t \equiv \mu_0 + \sqrt{2}\rho k_D^2 \Lambda^2 D$$
$$k = k_0 + k_t \equiv k_0 + \frac{\mu_t C_p}{\mathrm{Pr}}$$

where D is the effective deformation rate, k_D is a dimensionless constant, Λ is the element size and Pr is the turbulent Prandtl number. Besides the data defined in the constant material model, the constants k_D and Pr are required.

- **Smagorinsky-Lilly Subgrid-Scale model**

  The effective viscosity and the heat conductivity are defined as,

$$\mu = \mu_0 + \mu_t = \mu_0 + \sqrt{2}\rho \Lambda_s^2 D$$
$$k = k_0 + k_t = k_0 + \frac{\mu_t C_p}{\mathrm{Pr}}$$
$$\Lambda_s = \min\left(\kappa d, k_D V^{1/3}\right)$$

  where Λ_s is the mixing length for sub-grid scales, κ is the Von Karman constant (default is $\kappa = 0.41$), d is the distance to the closest wall and V is the element volume.

  The Smagorinsky-Lilly model can be used in conjunction with wall functions for near-wall modeling.

- **RNG-based Subgrid-Scale Model**

  The RNG-based effective subgrid viscosity is,

$$\mu = \mu_0 + \mu_t \equiv \mu_0 \left[1 + H(x)\right]^{1/3}$$

where

$$H(x) = \begin{cases} x, & x > 0 \\ 0, & x \le 0 \end{cases}$$
$$x = \mu_s^2 / \mu_0^2 - 100$$
$$\mu_s = \rho \sqrt{2} D \left(C_{RNG} \Delta\right)^2$$

in which, $C_{RNG} = 0.09348$ and Δ is the element size. The heat conductivity is defined in the same way as in the Smagorinsky-Lilly model.

The RNG-based Subgrid-Scale model is only available for FCBI-C elements.

When using the Smagorinsky-Lilly or the RNG-based subgrid-scale models, it is possible to superimpose time-dependent random fluctuations to the mean velocity value at the inlets.

3.6.9 User-supplied materials

In this model, the user can specify the material data through a user-supplied subroutine written in Fortran. It is designed only for incompressible, slightly compressible and low-speed compressible flows. The subroutine is invoked whenever material data are required in the computation. All necessary data is passed in as arguments, and the program returns the required material data. The arguments are the coordinates of the current spatial point, the time, the element group and element labels, current solution variables and their derivatives at the point, etc. The returned data are the same as in the constant model.

Using this model, the user can define many material data, for example, the mixing-length zeroth order turbulence model, where the viscosity is computed as

$$\mu = \mu_0 + \sqrt{2}\rho l^2 D$$

with μ_0 the laminar viscosity, ρ the density, l the mixing length and D the effective deformation rate.

As another example, the user can invoke their own subroutines or libraries to compute the material data, possibly in a very complicated way, say solving some partial differential equations.

The following sample example defines the material data for four different problems. In order to shorten the descriptions, we define a piecewise linear function as shown in the following figure.

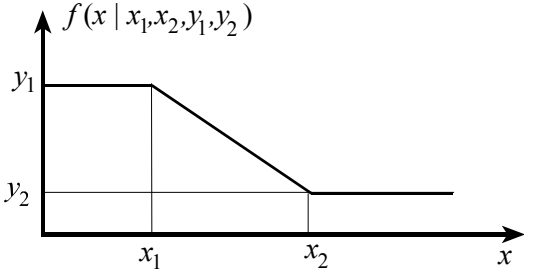


Figure 3.26 A piecewise linear function

(1) For the problem with the heading "CATERP…": The material is dependent of the mass-ratio ϕ_1. The original fluid is assumed to be air. The mass-ratio is used in simulating the ratio of the oil and the mixture, so that the fluid medium varies as air, air-oil mixture and oil, depending on the value of the mass-ratio.

$$\mu = f\left(\phi_1 \middle|\ 0,\ 1,\ 2.082\times10^{-5},\ 3.56\times10^{-2}\right)$$
$$\rho = f\left(\phi_1 \middle|\ 0,\ 1,\ 0.995,\ 853.9\right)$$
$$C_p = f\left(\phi_1 \middle|\ 0,\ 1,\ 1009,\ 2118\right)$$
$$k = f\left(\phi_1 \middle|\ 0,\ 1,\ 0.0304,\ 0.138\right)$$
$$\beta,\ \theta_0,\ g_1,\ g_2,\ g_3,\ \sigma,\ \kappa,\ C_v = 0$$

(2) For the problem with the heading "F7518…", the material is of varying density, which is assumed to be a function of temperature:

$$\rho = f\left(\theta \middle|\ 100,\ 1000,\ 0.086,\ 0.053\right)$$
$$\mu = 8\times10^{-6}$$
$$C_p = 14030$$
$$k = 0.156$$
$$\beta,\ \theta_0,\ g_1,\ g_2,\ g_3,\ \sigma,\ \kappa,\ C_v = 0$$

(3) For the problem with the heading "F7519…", the material density is assumed to be a function of the *z*-coordinate:

$$\rho = 1 + \frac{2}{5}z$$

and

$$\rho = f\left(z \middle|\ 0,\ Z,\ 1,\ 1+\tfrac{2}{5}Z\right) \qquad (Z = \text{const.} > z)$$
$$\mu = 0.2$$
$$C_p,\ k,\ \beta,\ \theta_0,\ g_1,\ g_2,\ g_3,\ \sigma,\ \kappa,\ C_v = 0$$

(4) For the problem with the heading "F74007…", the viscosity and density are defined as a function of the mass-ratio ϕ_1:

$$\mu = f\left(\phi_1 \middle| \ 0.1,\ 0.9,\ 0.1,\ 0.01\right)$$

$$\rho = f\left(\phi_1 \middle| \ 0.1,\ 0.9,\ 10,\ 1\right)$$

$$C_p,\ k,\ \beta,\ \theta_0,\ g_1,\ g_2,\ g_3,\ \sigma,\ \kappa,\ C_v = 0$$

```
      SUBROUTINE ADFUSR(CHEAD,L,NL,R,NR,CMAT,NCMAT)
C=======================================================================
C
C
C     USER-SUPPLIED MATERIAL DATA FOR INCOMPRESSIBLE FLOWS WITHOUT
C     TURBULENT-KE OR TURBULENT-KW MODELS
C
C     THIS ROUTINE WILL BE CALLED AT EACH INTEGRATION POINT OF EVERY
C     ELEMENT OF EVERY ELEMENT GROUP WHERE THE MATERIAL TYPE
C    "TURBULENT-MIXL" WAS SPECIFIED, AT EACH NONLINEAR EQUILIBRIUM
C     ITERATION AT EACH TIME STEP.
C
C     IN ORDER TO ACCESS THIS ROUTINE, USER MUST CHOOSE THE MATERIAL
C     TYPE "TURBULENT-MIXL".
C
C     USER MUST SPECIFY ALL MATERIAL DATA EXPLAINED HERE INTO THE ARRAY
C     CMAT(NCMAT) IN THIS ROUTINE. DEFINE ZERO VALUES IF THEY ARE
C     NOT USED.
C
C     WARNING: USER IS SUPPOSED TO BE AWARE OF THE AVAILABILITY OF THE
C              DATA STORED IN THE ARRAYS L(*) AND R(*) FOR THE
C              PROBLEM BEING SOLVED. IF NOT AVAILABLE, ARRAYS COULD
C              BE RANDOM.
C
C     ------------------------------------------------------------------
C
C     PASSED IN:  CHEAD, L(NL),R(NR)
C
C     PASSED OUT: CMAT(NCMAT)
C
C     ------------------------------------------------------------------
C     WHERE
C
C        CHEAD = THE STRING OF THE PROBLEM HEADING
C
C        L( 5) = IELG   = ELEMENT GROUP NUMBER
C        L( 6) = IELM   = ELEMENT NUMBER
C        L( 7) = IU     = ENTRY OF THE X-VELOCITY      IN R(*)
C        L( 8) = IV     = ENTRY OF THE Y-VELOCITY      IN R(*)
C        L( 9) = IW     = ENTRY OF THE Z-VELOCITY      IN R(*)
C        L(10) = IP     = ENTRY OF THE PRESSURE        IN R(*)
C        L(11) = IT     = ENTRY OF THE TEMPERATURE     IN R(*)
C        L(12) = IK     = ENTRY OF THE TURBULENCE-K    IN R(*)
C        L(13) = IE     = ENTRY OF THE TURBULENCE-E/W  IN R(*)
C        L(14) = IX     = ENTRY OF THE X-COORDINATE    IN R(*)
C        L(15) = IY     = ENTRY OF THE Y-COORDINATE    IN R(*)
C        L(16) = IZ     = ENTRY OF THE Z-COORDINATE    IN R(*)
C        L(19) = ITIM   = ENTRY OF THE TIME            IN R(*)
C        L(20) = IM     = ENTRY OF THE MASS-RATIOS     IN R(*)
C
C        R(IU)   = U    = X-VELOCITY
C        R(IU+1) = DXU  = DERIVATIVE OF U WITH RESPECT TO X
C        R(IU+2) = DYU  = DERIVATIVE OF U WITH RESPECT TO Y
C        R(IU+3) = DZU  = DERIVATIVE OF U WITH RESPECT TO Z
```

```
C        R(IV)   = V    = Y-VELOCITY
C        R(IV+1) = DXV  = DERIVATIVE OF V WITH RESPECT TO X
C        R(IV+2) = DYV  = DERIVATIVE OF V WITH RESPECT TO Y
C        R(IV+3) = DZV  = DERIVATIVE OF V WITH RESPECT TO Z
C        R(IW)   = W    = Z-VELOCITY
C        R(IW+1) = DXW  = DERIVATIVE OF W WITH RESPECT TO X
C        R(IW+2) = DYW  = DERIVATIVE OF W WITH RESPECT TO Y
C        R(IW+3) = DZW  = DERIVATIVE OF W WITH RESPECT TO Z
C        R(IP)   = P    = PRESSURE
C        R(IP+1) = DXP  = DERIVATIVE OF P WITH RESPECT TO X
C        R(IP+2) = DYP  = DERIVATIVE OF P WITH RESPECT TO Y
C        R(IP+3) = DZP  = DERIVATIVE OF P WITH RESPECT TO Z
C        R(IT)   = T    = TEMPERATURE
C        R(IT+1) = DXT  = DERIVATIVE OF T WITH RESPECT TO X
C        R(IT+2) = DYT  = DERIVATIVE OF T WITH RESPECT TO Y
C        R(IT+3) = DZT  = DERIVATIVE OF T WITH RESPECT TO Z
C        R(IK)   = TK   = TURBULENCE_K
C        R(IK+1) = DXK  = DERIVATIVE OF TK WITH RESPECT TO X
C        R(IK+2) = DYK  = DERIVATIVE OF TK WITH RESPECT TO Y
C        R(IK+3) = DZK  = DERIVATIVE OF TK WITH RESPECT TO Z
C        R(IE)   = TE   = TURBULENCE_E OR TURBULENCE-W
C        R(IE+1) = DXE  = DERIVATIVE OF TE WITH RESPECT TO X
C        R(IE+2) = DYE  = DERIVATIVE OF TE WITH RESPECT TO Y
C        R(IE+3) = DZE  = DERIVATIVE OF TE WITH RESPECT TO Z
C        R(IX)   = X    = X-COORDINATE
C        R(IY)   = Y    = Y-COORDINATE
C        R(IZ)   = Z    = Z-COORDINATE
C        R(ITIM) = TIME = TIME
C        R(IM)   = S    = MASS-RATIO-1
C        R(IM+1) = DXS  = DERIVATIVE OF S WITH RESPECT TO X
C        R(IM+2) = DYS  = DERIVATIVE OF S WITH RESPECT TO Y
C        R(IM+3) = DZS  = DERIVATIVE OF S WITH RESPECT TO Z
C        ...... CONTINUE FOR MASS-RATIO-2,...,UNTIL THE MASS-RATIO-10
C
C     ------------------------------------------------------------------
C
C        CMAT(1 ) = XMU   = VISCOSITY
C        CMAT(2 ) = CP    = SPECIFIC HEAT AT CONSTANT PRESSURE
C        CMAT(3 ) = XKCON = THERMAL CONDUCTIVITY
C        CMAT(4 ) = BETA  = COEFFICIENT OF VOLUME EXPANSION
C        CMAT(5 ) = RO    = DENSITY
C        CMAT(6 ) = TC    = REFERENCE TEMPERATURE
C        CMAT(7 ) = GRAVX = X-GRAVITATION
C        CMAT(8 ) = GRAVY = Y-GRAVITATION
C        CMAT(9 ) = GRAVZ = Z-GRAVITATION
C        CMAT(10) = QB    = RATE OF HEAT GENERATED PER VOLUME
C        CMAT(11) = SIGMA = COEFFICIENT OF SURFACE TENSION
C        CMAT(12) = XKAPA = BULK MODULUS
C        CMAT(13) = CV    = SPECIFIC HEAT AT CONSTANT VOLUME
C
C=======================================================================
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      CHARACTER*80 CHEAD
      DIMENSION L(NL),R(NR),CMAT(NCMAT)
C-----------------------------------------------------------------------
C     WRITE YOUR OWN CODE HERE AND GO TO 9999
C-----------------------------------------------------------------------
      IF(CHEAD(1:6).EQ.'CATERP') THEN
C
C        GET THE CURRENT VALUE OF MASS-RATIO
C
         IM       =  L(20)
         S        =  R(IM)
```

```
C
C        SET UP THE AIR MATERIALS
C
         DENAIR     =  0.995D0
         VISAIR     =  2.082D-5
         CPVAIR     =  1009.D0
         XKCAIR     =  0.0304D0
C
C        SET UP THE OIL MATERIALS
C
         DENOIL     =  853.9D0
         VISOIL     =  3.56D-2
         CPVOIL     =  2118.D0
         XKCOIL     =  0.138D0
C
C        COMPUTE THE VISCOSITY AND DENSITY OF THE MIXTURE
C
         IF(S.GT.1.D0) THEN
            VIS      =  VISOIL
            DEN      =  DENOIL
            CPV      =  CPVOIL
            XKC      =  XKCOIL
         ELSE IF(S.LE.0.D0) THEN
            VIS      =  VISAIR
            DEN      =  DENAIR
            CPV      =  CPVAIR
            XKC      =  XKCAIR
         ELSE
            VIS      =  VISAIR + S*(VISOIL-VISAIR)
            DEN      =  DENAIR + S*(DENOIL-DENAIR)
            CPV      =  CPVAIR + S*(CPVOIL-CPVAIR)
            XKC      =  XKCAIR + S*(XKCOIL-XKCAIR)
         ENDIF
C
C        PASS OUT ALL MATERIAL DATA REQUIRED
C
         CMAT(1 ) =  VIS
         CMAT(2 ) =  CPV
         CMAT(3 ) =  XKC
         CMAT(4 ) =  0.D0
         CMAT(5)  =  DEN
         CMAT(6 ) =  0.D0
         CMAT(7 ) =  0.D0
         CMAT(8 ) =  0.D0
         CMAT(9 ) =  0.D0
         CMAT(10) =  0.D0
         CMAT(11) =  0.D0
         CMAT(12) =  0.D0
         CMAT(13) =  0.D0
C
         GO TO 9999
      ENDIF
C
C-----------------------------------------------------------------------
      IF(CHEAD(1:6).EQ.'F71001' .OR. CHEAD(1:5).EQ.'F7518') THEN
C       +-------------------------------------------------+
C       | THIS FOR THE INTERNAL VERIFICATION PROBLEM F71001 |
C       | WHICH HAS A TEMPERATURE DEPENDENT DENSITY         |
C       +-------------------------------------------------+
         CMAT(1 ) =  8.D-6
         CMAT(2 ) =  14030.D0
         CMAT(3 ) =  0.156D0
         CMAT(4 ) =  0.D0
```

```
            CMAT(5 ) =  0.086D0
            CMAT(6 ) =  0.D0
            CMAT(7 ) =  0.D0
            CMAT(8 ) =  0.D0
            CMAT(9 ) =  0.D0
            CMAT(10) =  0.D0
            CMAT(11) =  0.D0
            CMAT(12) =  0.D0
            CMAT(13) =  0.D0
C
            IT       =  L(11)
            T        =  R(IT)
            TP1      =  100.D0
            TP2      =  1000.D0
            DP1      =  0.086D0
            DP2      =  0.053D0
            IF(T .LT. TP1) THEN
               CMAT(5) =  DP1
            ELSE IF(T .LT. TP2) THEN
               CMAT(5) =  DP1 + (T-TP1)*(DP2-DP1)/(TP2-TP1)
            ELSE
               CMAT(5) =  DP2
            ENDIF
         ELSE IF(CHEAD(1:5).EQ.'F7100' .OR. CHEAD(1:5).EQ.'F7519') THEN
C         +-------------------------------------------------+
C         | THIS FOR THE INTERNAL VERIFICATION PROBLEM F7100X |
C         | WHICH HAS A Z-COODINATE DEPENDENT DENSITY (X=3-8) |
C         +-------------------------------------------------+
            CMAT(1 ) =  0.2D0
            CMAT(2 ) =  0.D0
            CMAT(3 ) =  0.D0
            CMAT(4 ) =  0.D0
            CMAT(5 ) =  0.D0
            CMAT(6 ) =  0.D0
            CMAT(7 ) =  0.D0
            CMAT(8 ) =  0.D0
            CMAT(9 ) =  0.D0
            CMAT(10) =  0.D0
            CMAT(11) =  0.D0
            CMAT(12) =  0.D0
            CMAT(13) =  0.D0
C
            IZ       =  L(16)
            Z        =  R(IZ)
            CMAT(5)  =  1.D0 + Z/10.D0  * 4.0D0
         ELSE IF(CHEAD(1:6).EQ.'F74007') THEN
C         +-----------------------------------------------------+
C         | THIS FOR THE INTERNAL VERIFICATION PROBLEM F74007   |
C         | WHICH HAS A MASS DEPENDENT DENSITY                  |
C         +-----------------------------------------------------+
            IM       =  L(20)
            S        =  R(IM)
            CMAT(1 ) =  0.01D0
            CMAT(2 ) =  0.D0
            CMAT(3 ) =  0.D0
            CMAT(4 ) =  0.D0
            CMAT(5 ) =  0.D0
            CMAT(6 ) =  0.D0
            CMAT(7 ) =  0.D0
            CMAT(8 ) =  0.D0
            CMAT(9 ) =  0.D0
            CMAT(10) =  0.D0
            CMAT(11) =  0.D0
```

```
         CMAT(12) =  0.D0
         CMAT(5)  =  1.D0
C
         IF(S.GT.0.9D0) THEN
            CMAT(1)  =  0.1D0
            CMAT(5)  =  10.D0
         ELSE IF(S.LT.0.1D0) THEN
            CMAT(1)  =  0.01D0
            CMAT(5)  =  1.D0
         ELSE
            RRR      = (S-0.1D0)/0.8D0
            CMAT(1)  =  0.01D0 + RRR*(0.1D0-0.01D0)
            CMAT(5)  =  1.D0   + RRR*(10.D0-1.D0  )
         ENDIF
      ENDIF
C-----------------------------------------------------------------------
 9999 CONTINUE
      RETURN
      END
```

3.6.10 ASME steam table

This model automatically calculates water and steam properties based on the ASME steam table (ASME International Steam Tables for Industrial Use, ASME Press, 2000). The user has the choice of whether to consider or not meta-stable vapor status. The following parameters and conversion factors are input by the user:

$\mathbf{g}$ = gravitational acceleration vector

q^B = rate of heat generated per unit volume

θ_0 = reference temperature

p_0 = reference pressure

$p_0^c, \Delta p^c$ = conversion factors for pressure, $p^m = \left(p^s - p_0^c \right) \big/ \Delta p^c$

$\theta_0^c, \Delta \theta^c$ = conversion factors for temperature, $\theta^m = \left(\theta^s - \theta_0^c \right) \big/ \Delta \theta^c$

$\Delta \rho^c$ = conversion factor for density, $\rho^m = \rho^s \big/ \Delta \rho^c$

ΔL^c = conversion factor for length, $L^m = L^s \big/ \Delta L^c$

where p^m, θ^m, ρ^m and L^m are the pressure, temperature, density and length specified in the model. Additionally, p^s, θ^s, ρ^s and L^m are the pressure (Pascal), temperature (K), density (kg/m^3) and length (m) in SI units as defined in the ASME stem tables.

In the incompressible flow model, ρ is the value of density at pressure p_0 and temperature θ_0. In the slightly compressible flow model, ρ^m is the value of density at pressure p and temperature θ. Furthermore, in low-speed compressible flow model, both ρ^m and ρ are the values of density at pressure p and temperature θ.

3.6.11 P-T-dependent material

When compared with the constant material model, this model has one additional parameter, namely, the reference pressure, p_0. In addition, the following parameters are a function of both pressure and temperature

$$\mu, C_p, C_v, k, \beta, q^B, \sigma, \kappa, \rho = f\left(p, \theta\right)$$

In the incompressible flow model, ρ is the density at pressure p_0 and temperature θ_0. In the slightly compressible flow model, $\rho_m = \rho\left(1+\left(p-p_0\right)/\kappa\right)$ and ρ is the density at pressure p_0 and temperature θ_0. Furthermore, in low-speed compressible flow model, both ρ_m and ρ are the density at pressure p and temperature θ.

3.6.12 Field friction

This model adds a field friction force to the right-hand side of the momentum equations. The force has the following form,

$$\mathbf{f}^B = -\frac{2f}{D}\rho\left\|\mathbf{v}\right\|\mathbf{C}\cdot\mathbf{v}$$

where $f = a\left(\mathrm{Re}_D\right)^b$, $\mathrm{Re}_D = \rho\left\|\mathbf{v}\right\|D/\mu$, $\mathbf{C} = C_i\mathbf{e}^i\mathbf{e}_i$, $\mathbf{e}_i, \left(i=1,2,3\right)$ are normalized independent direction vectors $\left((\mathbf{e}_1\times\mathbf{e}_2)\cdot\mathbf{e}_3 \neq 0\right)$, $\rho, \mathbf{v}, \mu$ are

local density, velocity, and viscosity respectively, D is characteristic length, and a, b are two dimensionless constants.

The parameters $D, a, b, C_1, C_2, C_3, \mathbf{e}_1, \mathbf{e}_2$ and $\mathbf{e}_3$ are required.

Note: In 2D models, $\mathbf{e}_1$ and the x- components of $\mathbf{e}_2$ and $\mathbf{e}_3$ are ignored.

3.6.13 User-modified materials

For incompressible, slightly compressible and low-speed compressible flows, the user can modify the material properties in the user-supplied subroutine USRMAT. This option covers all type of materials described above except the user-supplied materials. In particular, it covers turbulence and porous media materials. The table at the beginning of the user subroutine lists all material properties that can be modified.

This subroutine is invoked whenever material data is required during computation. All necessary data is passed in through arguments and the program returns the material data that has been modified. User may modify part or all of the material data. For unchanged data, the values defined in the model material data are used.

If a field friction loading is applied to the element group, user may also modify it in this subroutine. Please note, once changed, the complete field friction data associated with this element group must be re-defined. They replace original friction data.

To activate this option, choose "Modify Material Properties During Solution with User Subroutine" in element group.

3.6.14 Material curves

A more general material set can be defined with the selected materials associated with up to three material curves of type “tabulated”, “polynomial” and “2-argument” respectively. Those material curves replace the default material data in the selected materials.

The following types of material model can be associated with the material curves:

Constant material model
Power-law model
Temperature-dependent power-law model
Carreau model
Second-order model
Large-eddy-simulation model
User-suuplied materials
Turbulent K-ε model
Turbulent K-ω model
Turbulent SA model
Porous material model (for flow through porous media)

The three material curves are introduces here after.

A *tabulated* material curve defines tabulated material data in the form

$$x_i : P_{i1}\left(x_i\right), P_{i2}\left(x_i\right), ..., P_{i,J}\left(x_j\right) \quad \left(i = 1, 2, ..., I\right)$$

where x can be time, temperature or pressure, and P are the values of the parameters to be modified. The possible parameters that can be selected are

ρ = density
μ = viscosity
C_p = specific heat at constant pressure
C_v = specific heat at constant volume
k = heat conductivity
q^B = rate of heat generated per unit volume
β = thermal expansion coefficient
$\mathbf{g}$ = gravitational acceleration vector
c_μ = the empirical constant in K-ε models
α = the empirical constant in K-ω models
κ = fluid bulk modulus of elasticity

A *polynomial* material curve defines material data in a polynomial

$$P_j = \sum_{i=1}^{I_i} a_i x^{e_i} \quad (j = 1, 2, ..., J)$$

where x and P are the same as defined for tabulated material curve.

A 2-argunment material curve defines material data varying as function of two arguments (any set of two arguments of time, temperature and pressure). The curve is input in the form

$$(x_i, MCV_i) \quad (i = 1, 2, ..., I)$$

where x can be time, temperature or pressure, and MCV_i are either tabulated or polynomial material curves. Thus, at x_i, the material values are determined by a material curve that is a function of another varying argument different from x. For example, if temperature represents x in the 2-argument material curve and, in which, MCV_i are pressure-dependent material curves of C_p and C_v, the curve defines temperature-pressure-dependent material parameters $C_p(\theta, p)$ and $C_v(\theta, p)$.

3.6.15 Gas models

The following state equations are available for low-speed compressible flows with FCBI-C elements associated with the total energy equation: ideal gas, standard Redlich-Kwong, Aungier-Redlich-Kwong, Soave-Redlich-Kwong, and Peng-Robinson. These are defined using the STATE-EQUATION command, and are applied using the STATE-EQ parameter of the MATERIAL commands. Different materials in the model can use different equations of state.

The state equations implicitly express the specific volume $v\left(=1/\rho\right)$ as a function of pressure, p, temperature, θ, and the specific gas constant, R_{sp}. Depending on the model, the critical pressure, p_C, critical temperature, θ_C, critical specific volume, v_C, and an acentric factor, ω, may also be required.

The state equation affects the internal energy of the fluid. This effect can be accounted in the energy equation using the concept of an "effective" specific heat at constant volume $\left(C_v^*\right)$, where $de = C_v^* d\theta$ and e is the specific internal energy. The effect can be switched on or off using the CV-EFFECTIVE parameter of the STATE-EQUATION command. If the effect is accounted for, C_v^* is different from the user-input value of C_v. If it is not accounted for, the user-input value of C_v is directly used in the energy equation (i.e., $C_v^* = C_v$ is used in this case).

In case that C_v is temperature-dependent (it is in most cases), define it by appllying a material curve in the selected material.

Ideal Gas law (IDEAL)

The ideal gas law is represented as

$$pv = R_{sp}\theta$$

where the specific gas constant, R_{sp}, is a user-defined input.

This gas model is used when C_v should vary in the energy equation, but the specific gas constant, R_{sp}, should not vary in the equation of state. In this model, the user-input C_v defined in the MATERIAL command is only used in the energy equation, and the user-input C_p is ignored.

Note that if STATE-EQ=0 in the MATERIAL command, the ideal gas law is used, where the specific gas constant is computed from the specific heats, $R_{sp} = C_p - C_v$.

Redlich Kwong models (RK, ARK, and SRK)

All Redlich-Kwong models can be expressed in the following general form

$$p = \frac{R_{sp}\theta}{v - b + c} - \frac{a}{v\left(v + b\right)}$$

where the expressions for a, b, and c depend on the type of Redlich-Kwong model used.

Standard Redlich Kwong model (RK)

In the standard Redlich-Kwong model, the expressions for a, b, and c are given by

$$a = a_0 \left(\frac{\theta}{\theta_C} \right)^{-n}$$

$$b = 0.08664 \frac{R_{sp}\theta_C}{p_C}$$

$$a_0 = 0.42747 \frac{R_{sp}^2 \theta_C^2}{p_C}$$

$$c = 0$$

$$n = \tfrac{1}{2}$$

where the required input parameters are R_{sp}, p_C, θ_C.

Aungier Redlich Kwong model (ARK)

The Aungier-Redlich-Kwong model is a modified version of the standard Redlich-Kwong model with

$$c = \frac{R_{sp}\theta_C}{p_C + \dfrac{a_0}{v_C\left(v_C + b\right)}} + b - v_C$$

$$n = 0.4986 + 1.1735\omega + 0.4754\omega^2$$

The required input parameters are $R_{sp}, p_C, \theta_C, \omega, v_C$.

Soave Redlich Kwong model (SRK)

The Soave-Redlich-Kwong model is another modified version of the standard Redlich-Kwong model with

$$a = a_0\left(1 + n\left(1 - \sqrt{\frac{\theta}{\theta_C}}\right)\right)^2$$

$$n = 0.480 + 1.574\omega - 0.176\omega^2$$

The required input parameters are $R_{sp}, p_C, \theta_C, \omega$.

Peng Robison model (PR)

The Peng-Robison model is expressed as

$$p = \frac{R_{sp}T}{v - b} - \frac{a}{v^2 + 2vb - b^2}$$

where a has the same expression as Soave-Redlich-Kwong model and

$$b = 0.0778 \frac{R_{sp}\theta_C}{p_C}$$

$$a_0 = 0.45724 \frac{R_{sp}^2\theta_C^2}{p_C}$$

$$n = 0.37464 + 1.54226\omega - 0.26993\omega^2$$

and the required input parameters are $R_{sp}, p_C, \theta_C, \omega$.

3.6.16 Conditional loads

A conditional load is a type of loading that is only active when certain conditions are satisfied. Currently, ADINA-F supports the following conditional loads for FCBI-C elements:

(1) Flow resistance loading
(2) Diffusion loading

The following conditions are available

(1) Boundary distance condition
(2) Outer iteration condition

Flow Resistance load

Flow resistance loading adds a field friction force to the fluid flow field. The force per unit volume has the following form

$$\mathbf{f} = -c\,\tfrac{1}{2} f \frac{\rho}{R_h} \|\mathbf{v}\| \mathbf{v}$$

where, ρ is the density, $\mathbf{v}$ is the velocity, R_h is the hydraulic radius, f is the friction coefficient, and c is a calibration factor. The friction coefficient is expressed as a function of local Reynolds number

$$f = \frac{a}{\left(\mathrm{Re}_D\right)^b}$$

where a and b are the two dimensionless empirical constants, Re_D is the Reynolds number based on the hydraulic diameter $D\left(\equiv 4R_h\right)$. The hydraulic radius is defined as the ratio of the channel cross-section area and the wetted parameter $R_h = A/W$.

For flow fields in fixed-wall channels, these parameters can be obtained either from analytical solutions or from experimental data under similar conditions. A few examples are shown below:

For laminar flows between two infinite parallel plates, $a = 24, b = 1$ and $R_h = \frac{1}{2}d$ can be obtained, where d is the distance between the two plates. These are the default settings for flow resistance loading.

For laminar flows in circular pipe, $a = 16, b = 1$ and $R_h = \frac{1}{2}R$ can be obtained, where R is the radius of the pipe. For turbulent flows, $a = 0.078, b = 0.25$ and $a = 0.046, b = 0.2$ can be obtained for rough and smooth pipes, respectively.

In moving-boundary channels or for FSI problems, the distance d varies depending on the solution time, and so does the hydraulic radius. ADINA-F can automatically calculate the distance, d, from which the hydraulic radius is determined using the two walls defined by the boundary distance condition.

If the moving wall motion is known, the hydraulic radius might also be known. In this case, R_h can be directly specified using an appropriate time function. The boundary distance condition is then not required, therefore, reducing solution time.

Flow resistance loading with a boundary distance condition can be used to model flow stoppage due to solid-to-solid contact. It can also be used to model flow leakage that might occur due to imperfect solid-to-solid contact

in the physical parts, for example, due to machining tolerances. This feature is particularly useful when the contact location continuously moves, or when the contact location depends on the FSI solution. Such applications include:

- Scroll compressors.
- Reciprocating piston compressors.
- Reed valves.
- Gear pumps.
- Peristaltic pumps.

The pressure difference and flow leakage between the boundaries depends on the amount of flow resistance loading applied. The user should calibrate the flow resistance loading coefficients c to match the pressure difference (or flow leakage) measured from experiment. Sometimes, the exponent, b, might also have to be calibrated.

The higher the flow resistance calibration coefficient, c, the smaller the flow leakage between the boundaries. However, if the flow resistance coefficient is excessively large, convergence will be slow or the solution will not converge. Indeed, if the flow resistance loading is infinite, the flow condition becomes unstable leading to divergence.

The flow resistance load feature has an option to convert the wall boundary condition to slip walls in the domain where the flow resistance load is applied. This option is specified using the WALL=SLIP parameter. When this option is used, the shear-stress on the walls is removed and 1D-model flow conditions can be established.

Diffusion load

Diffusion loading adds additional diffusion to the fluid flow field by adding viscosity and heat conductivity to the originally input viscosity and heat conductivity.

When diffusion loading is used with the boundary distance condition in turbulence models, “laminar-like” flow is assumed when the boundary distance condition is closed. This is known as conditional turbulence. Conditional turbulence can improve the convergence of solution for

problems with thin gaps, where the flow is expected to be physically laminar in the gap.

Diffusion loading can be used as an alternative tool in modeling the problems where the flow resistance loading can be applied.

Boundary-Distance condition

The boundary distance condition scales the load depending on the distance between two boundaries. In this condition, the following input data is required:

- Two boundaries.
- Two critical boundary-distance values d_{closed} and d_{open}.

The boundaries must be external solid boundaries of the fluid domain (fixed wall, moving wall, or fluid-structure interface). Each boundary surface must be continuous, and the two boundary surfaces cannot be connected.

For each element, the distance between the two boundaries, d, is computed. This distance is defined as sum of the shortest distance from the element centroid to each boundary, $d = d_1 + d_2$. The load is fully applied to the element when $d \le d_{closed}$, and no load is applied when $d > d_{open}$. When d varies from d_{closed} to d_{open}, the load is linearly decreased. As an option, the shear-stress on the walls may be removed when $d < d_{closed}$, to model a 1D channel flow.

In order to smoothly apply the load, the open distance, d_{open}, should be larger than the closed distance, d_{closed}. This improves the convergence rate, as it prevents the flow conditions from abruptly changing in the region where the load is applied.

Outer-Iteration condition

The outer-iteration condition defines when to apply a load during solution iterations. In this condition, the following input data is required:

- Time range t_{start} and t_{end}.
- Two iteration stages i_{end} and Δi_{end}.

Under this condition, the load is fully applied when the simulation time t is in the range $t_{start} \le t \le t_{end}$, and the iteration number is less than i_{end}. After this iteration, the load linearly decreases and eventually diminishes after Δi_{end} iterations. Obviously, $t_{start} \le t_{end}$ and $0 \le i_{end}, \Delta i_{end}$ must be satisfied.

One application of the outer-iteration condition is to improve the convergence of solution. In this application, i_{end} should not be too large (typically, less than 100 for difficult problems), otherwise, false solution could be obtained. Normally, Δi_{end} should be a few iterations (around 10).

As an option, ADINA-F may automatically reduce i_{end} if the solution with the load applied converges before the user-specified value of i_{end}. With this option, the converged solution is always free of the applied loading. If the option is not used, the converged solution is only free of the applied loading if the program has performed more than $i_{end} + \Delta i_{end}$ outer iterations.

This page intentionally left blank

Chapter 4 Formulation of fluid flows through porous media

In this chapter, we introduce our formulation for fluid flows and heat transfer in porous media.

The model introduced here can be coupled with ADINA solid models in fluid-structure interaction analyses. It can also be associated with mass transfer analyses. Used as additional element groups, it can also be employed in incompressible, slightly compressible and low-speed compressible flow solutions.

The following table presents a quick overview of the capabilities that are available for porous media flows.

Table 4-1 Functionalities in the porous media flow model

Category	Functionality	
Coupled models	ADINA solid model	
	incompressible, slightly compressible and low-speed compressible flows	
	solid element groups	
	heat transfer	
	mass transfer	
	electro-static and steady current conduction analyses	
	volume of fraction	
	liquid-vapor phase change	
Computational domains	2D planar, 2D axisymmetric and 3D	
Analyses	steady-state and transient	
Galerkin Elements	2D: 3-node linear element and 6/9-node bilinear elements	
	3D: 4-node linear element and 27-node bilinear elements	
FCBI and FCBI-C elements	2D: 3/4-node linear elements	
	3D: 4/5/6/8-node linear elements	
Material models	constant	
Boundary conditions Note: The column furthest to the right represents the solution variables or equations that are immediately affected by the	prescribed velocities	**v**
	zero velocities	
	prescribed pressure	*p*
	zero pressure	
	prescribed rotational velocity	**v**
	concentrated force load	
	distributed normal-traction load	

affected by the specified condition. The solution variable **d** indicates a moving boundary condition.	field centrifugal load	
	fixed wall	
	uniform flow	
	angular velocity	
	moving wall	**v, d**
	fluid-structure interface	
	free surface	**d**
	fluid-fluid interface	
	phase-change	
	gap	all
	prescribed temperature	θ
	zero temperature	
	concentrated heat flow load	
	distributed heat flux load	
	convection	
	radiation	
	specular radiation	
	thermal resistance	
	user-supplied	all
Initial conditions	zero (default conditions)	
	specified	
	mapped from other solutions	
	restart run	
Solvers for linearized equations	Gauss elimination method (COLSOL)	
	sparse solver	
	iterative methods (RPBCG, RPGMRES, and AMG solvers)	
Other capabilities	automatically nondimensional procedure	
	automatic time-stepping CFL option	
	automatic time-stepping ATS option	
	skew system	
	constraint condition	
	conjugate heat transfer	
	element birth-death option	
	pressure datum	
	include/exclude hydrostatic pressure	
	physical or mathematical formulations for incompressible flows	
	with or without dissipations	

4.1 Governing equations

The governing equations used in our formulations are the continuity equation, energy equation and Darcy's equation (momentum equation). They are listed in Table 2-1. The solution variables are the primitive

variables $(p, \mathbf{v}, \theta)$. In case of moving mesh problems, the ALE formulation described in Section 2.14 is used.

4.2 Numerical method

The numerical method for continuity and energy equations in porous media is the same as for incompressible flow analyses.

The momentum equation is weighted with the virtual quantity of velocity and is integrated over the computational domain V. The divergence theorem is used to lower the order of the derivatives of the pressure, resulting in the expression of natural boundary conditions. This natural condition is part of usual boundary conditions. The variational form is written as

$$\int_V \left[h^{\mathrm{v}} (\mu \boldsymbol{\kappa}^{-1} \bullet \mathbf{v} - \mathbf{f}^B) - p \nabla h^{\mathrm{v}} \right] dV = - \oint h^{\mathrm{v}} p d\mathbf{S} \qquad (1.1)$$

where h^{v} the is virtual quantity of velocity. The associated elements can be any one that is applicable to porous media flows. Upwinding techniques are not applied to Eq.(1.1).

In 2D 4-node and 3D 8-node elements, a finite volume method is used. Using the related interpolation functions described in Chapter 10, the discretized equations are

$$\int_V \left[\mu \boldsymbol{\kappa}^{-1} \bullet (\mathbf{v} - \mathbf{w}) - \mathbf{f}^B \right] dV + \int_S p d\mathbf{S} = 0$$

Similar to incompressible flows, all variables at every node will form the solution vector that is governed by the finite element equation. Both the Newton-Raphson and successive substitution methods are available to solving the nonlinear equations. All solvers introduced in Chapter 11 are applicable to solving the linear algebraic equations.

The same solution procedure as for solving incompressible flows is carried out for solving porous media flow problems.

4.3 Elements

The elements that can be used for porous media are the same as those for incompressible flows.

4.4 Boundary conditions

Using proper boundary conditions is essential to successfully solving fluid flow problems. The required boundary conditions strongly depend on the type of flows or, equivalently, on the governing equations used and the associated computational domain.

Fluid velocity and pressure are always strongly coupled and must be treated like one variable as far as boundary conditions are concerned. On the other hand, the fluid flow might not be directly coupled to the temperature (although the flow may be coupled to the temperature field through temperature-dependent materials and the buoyancy force in the Boussinesq approximation). Therefore, as a general rule, they must be considered separately when applying boundary conditions.

A typical fluid domain and associated boundary conditions are shown in the following figure, where the whole boundary S of the computational domain has been divided into $\overline{S}_1$ (where velocities are prescribed) and $\overline{S}_2$ (where a normal-traction load is applied). The S_i here represents the interior part of that part of the boundary, while the over bar indicates that it contains the boundary of S_i as well. Therefore, the common part of S_1 and S_2 is the empty geometric set, while $\overline{S_1}$ and $\overline{S_2}$ share their common frontier where both conditions are applied. This boundary partition can be expressed as

$$\overline{S}_1 \cup \overline{S}_2 = \overline{S} = S, \quad S_1 \cap S_2 = \varnothing$$

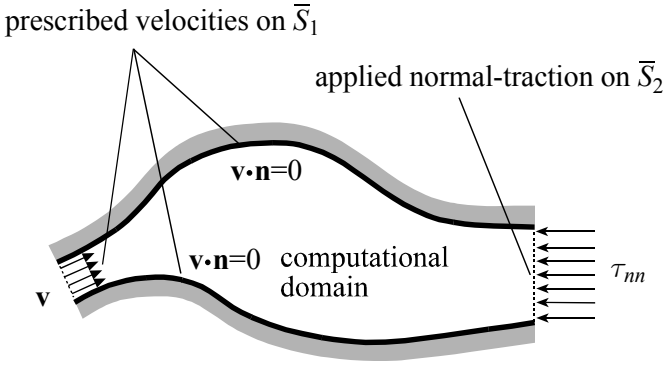


Figure 4.1 A typical boundary condition set for porous media flows

From this set of conditions, we observe that:

- All parts of the boundary have been assigned one and only one fluid boundary condition (except on the interface of the sub-boundaries, where two conditions meet).
- For steady-state analyses, at least one velocity condition and one pressure condition must be specified (the distributed normal- traction load acts as a prescribed pressure condition).
- A slip condition is usually applied on walls.

These observations are fundamental guidelines for specifying boundary conditions in most porous media flow problems.

The discussion of boundary conditions for heat transfer in Section 3.4 is also valid for heat transfer in porous media.

All the boundary conditions available in incompressible flows are also applicable to porous media flows. These conditions have been introduced in Section 3.4.

4.5 Initial conditions

In transient analyses, all solution variables must be specified. The default initial conditions to all variables are zeros. Although initial conditions are not required in steady-state analyses, they are used as a "guessed" solution at the start of the equilibrium iterations. A good initial condition may accelerate the convergence during equilibrium iterations. In certain cases, the initial condition may become a key factor in obtaining converged solutions.

Initial conditions can be improved in restart analyses or using the solutions mapped from a similar solution.

4.6 Material model

The following data are required in porous media flow models:

ρ = fluid density
ρ_s = solid density
μ = fluid viscosity
$\mathbf{g}$ = gravitational acceleration vector
ϕ = porosity
$\boldsymbol{\kappa}$ = permeability tensor
C_{p_f} = specific heat of fluid
C_{p_s} = specific heat of solid
k_f = thermal conductivity of fluid
k_s = thermal conductivity of solid
q^B = rate of heat generated per unit volume
β = thermal expansion coefficient
θ_0 = reference temperature in buoyancy force
σ = coefficient of surface tension
κ = bulk modulus of elasticity

Note: A more general time, temperature or pressure dependent porous material set can be defined using material curves. See corresponding sections in Chapter 3 for more details.

This page intentionally left blank

Chapter 5 Formulation of turbulence in incompressible, slightly compressible and low-speed compressible flows

In this chapter, we introduce our two-equation turbulence models for incompressible and slightly compressible flows with heat transfer. These models are also applicable to low-speed compressible flows.

The models introduced here can be coupled with ADINA solid models in fluid-structure interaction analyses. They can also be used in mass transfer analyses.

The following table presents a quick overview of the capabilities that are available for turbulence models for incompressible, slightly compressible and low-speed compressible flows.

Table 5-1 Functionalities in K-ε and K-ω turbulence models for incompressible, slightly compressible and low-speed compressible flows

Category	Functionality
Coupled models Note: Heat transfer must be coupled in low-speed compressible flow models.	ADINA solid model
	solid element groups
	heat transfer
	mass transfer
	Electro-static and steady current conduction analyses
	liquid-vapor phase change
Computational domains	2D planar, 2D axisymmetric and 3D
Analyses	steady-state and transient
Galerkin Elements	2D: 3-node linear element
	3D: 4-node linear element
FCBI and FCBI-C elements	2D: 3/4-node linear elements
	3D: 4/5/6/8-node linear elements
Material models	sandard turbulent K-ε and K-ε RNG
	high and low Reynolds turbulent K-ω
	Shear Stress Transport (SST) model

Boundary conditions Note: The column furthest to the right represents the solution variables or equations that are immediately affected by the specified condition. The solution variable **d** indicates a moving boundary condition.	Prescribed velocities	**v**
	zero velocities	
	prescribed pressure	p
	zero pressure	
	prescribed rotational velocity	**v**
	concentrated force load	
	distributed normal-traction load	
	field centrifugal load	
	fixed wall	$\mathbf{v}, K, \varepsilon/\omega$
	uniform flow	**v**
	angular velocity	
	moving wall	$\mathbf{v}, K, \varepsilon/\omega, \mathbf{d}$
	fluid-structure interface	
	free surface	**d**
	fluid-fluid interface	
	phase-change	
	Gap	all
	prescribed temperature	θ
	zero temperature	
	concentrated heat flow load	
	distributed heat flux load	
	convection	
	radiation	
	specular radiation	
	thermal resistance	
	user-supplied	all
	prescribed turbulence K-ε/ω	$K, \varepsilon/\omega$
	zero flux of turbulence variables	$K, \varepsilon/\omega$
Initial conditions	zero (default conditions)	
	specified	
	mapped from other solutions	
	restart run	
Solvers for linearized equations	Gauss elimination method (COLSOL)	
	sparse solver	
	iterative methods (RPBCG, RPGMRES, and AMG solvers)	
Other capabilities	automatically nondimensional procedure	
	automatic time-stepping CFL option	
	automatic time-stepping ATS option	
	skew system	
	constraint condition	
	conjugate heat transfer	
	element birth-death option	
	pressure datum	
	include/exclude hydrostatic pressure	
	with or without dissipations	

5.1 Governing equations

The governing equations for the fluid flow are those of incompressible, slightly compressible or low-speed compressible flows. However, the viscosity and heat conductivity are modified to include the turbulence effect and the K-ε and K-ω models are used to provide the required variables. These equations are listed in Table 2-1. The solution variables are $(p, \mathbf{v}, \theta, K, \varepsilon)$ in the K-ε models and $(p, \mathbf{v}, \theta, K, \omega)$ in the K-ω models. In the case of moving mesh problems, the ALE formulation described in Section 2.14 is used.

In the K-ε models, both the standard and the RNG models can be used. In the K-ω models, both the high and low Reynolds models can be used.

5.2 Numerical method

The time integration is the same as for the fluid flow equations. The fluid equations and their discretization are described in Section 3.2.

The finite element method is used to discretize the governing equations of the turbulence. As for the fluid flow equations, the finite element equations are obtained by establishing a weak form of the turbulence equations using the Galerkin procedure. The K, ε and ω equations are weighted with the virtual quantities of K, ε and ω, respectively. The equations are integrated over the computational domain V. The divergence theorem is used to lower the order of the derivatives of turbulence fluxes.

The variational forms for the K-ε equations are

$$\int_V \left(h^f G^f + \mathbf{Q}^f \bullet \nabla h^f \right) dV = 0$$

where f represents K and ε, h^K and h^ε are virtual quantities of K and ε respectively, and

$$G^K = y^a \left(\rho \frac{\partial K}{\partial t} + \rho \mathbf{v} \bullet \nabla K - S_K \right)$$

$$G^\varepsilon = y^a \left(\rho \frac{\partial \varepsilon}{\partial t} + \rho \mathbf{v} \bullet \nabla \varepsilon - S_\varepsilon \right)$$

$$\mathbf{Q}^K = y^a \mathbf{q}_K$$

$$\mathbf{Q}^\varepsilon = y^a \mathbf{q}_\varepsilon$$

Similarly, finite element equations for the K-ω models are obtained with f representing K and ω, and h^K and h^ω are virtual quantities of K and ω, respectively, and

$$G^K = y^a \left(\rho \frac{\partial K}{\partial t} + \rho \mathbf{v} \bullet \tilde{\mathrm{N}} K - G_K \right)$$

$$G^\omega = y^a \left(\rho \frac{\partial \omega}{\partial t} + \rho \mathbf{v} \bullet \tilde{\mathrm{N}} \omega - G_\omega \right)$$

$$\mathbf{Q}^K = y^a \mathbf{q}_K$$

$$\mathbf{Q}^\omega = y^a \mathbf{q}_\omega$$

The upwinding techniques are also used in constructing the finite element equations governing the turbulence variables.

In FCBI and FCBI-C elements, conservative forms associated with a finite volume method is used, in which the corresponding terms are defined as

$$\mathbf{f} = y^a \begin{bmatrix} \rho K \\ \rho \varepsilon \\ \rho \omega \end{bmatrix}, \quad \mathbf{B} = y^a \begin{bmatrix} \rho \mathbf{v} K - \mathbf{q}_K \\ \rho \mathbf{v} \varepsilon - \mathbf{q}_\varepsilon \\ \rho \mathbf{v} \omega - \mathbf{q}_\omega \end{bmatrix}, \quad \mathbf{R} = y^a \begin{bmatrix} G_K \\ G_\varepsilon \\ G_\omega \end{bmatrix}$$

5.3 Elements

The elements that can be used for turbulence flows here are the same as those used for incompressible flows except that bilinear elements are not applicable.

5.4 Boundary conditions

The boundary conditions for fluid variables, $(\mathbf{v}, p, \theta)$, are usually the same as for laminar models, although the treatment of the wall condition or its analogue is modified as explained later in this section. The conditions introduced here are only for the turbulence variables K, ε and ω.

A typical fluid domain and its associated boundary conditions for turbulence variables, K, ε and ω, are shown in the following figure, where the whole boundary S of the computational domain has been divided into $\overline{S}_5$ (where the turbulence variables are prescribed), $\overline{S}_6$ (where zero turbulence fluxes are applied) and $\overline{S}_7$ (where wall conditions have been applied). The S_i here represents the interior part of that part of the boundary, while the over bar indicates that it contains the boundary of S_i as well. Therefore, the common part of S_i and S_j is the empty geometric set when $i \neq j$, while $\overline{S_i}$ and $\overline{S_j}$ share their common frontier where both conditions are applied. This boundary partition can be expressed as

$$\overline{S}_5 \cup \overline{S}_6 \cup \overline{S}_7 = \overline{S} = S$$
$$S_i \cap S_j = \varnothing \qquad i \neq j, \quad i, j = 5,6,7$$

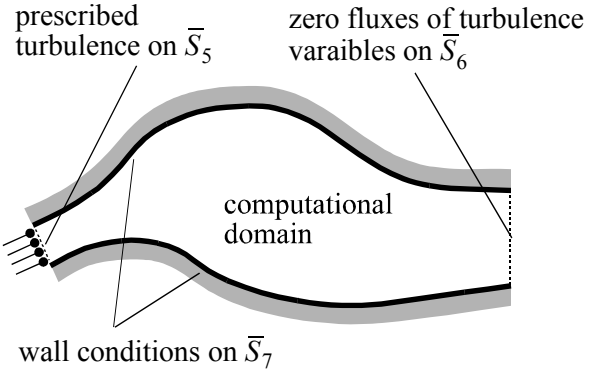


Figure 5.1 A typical boundary condition set for turbulence variables

From this set of conditions, we observe that:

- All parts of the boundary have been assigned one and only one turbulence boundary condition (except on the interface of the sub-boundaries, where two conditions meet).
- There is at least one wall condition or a prescribed turbulence variable condition.

These observations are fundamental guidelines for specifying boundary conditions for turbulence variables. The available conditions are introduced here.

5.4.1 Prescribed turbulence K-ε

prescribed turbulence K-ε

This condition can only be applied to a K-ε model. Time-dependent turbulence variables are directly prescribed

$$K = \overline{K}(t)$$
$$\varepsilon = \overline{\varepsilon}(t)$$

and applied to boundaries. The K and ε equations at boundary nodes are then replaced by these equations.

In addition to directly given values, they can also be prescribed indirectly via

$$K = \tfrac{3}{2}\left(\bar{i}\,\bar{v}\right)^2$$
$$\varepsilon = K^{\frac{3}{2}} \Big/ \left(0.3\bar{L}\right)$$

where, $\bar{v}, \bar{L}$ and $\bar{i}$ $(= 0.01 \sim 0.1)$ are velocity scale, length scale and turbulence intensity respectively.

Note that, in SA and DES models, through all input and output, K represents $\tilde{\nu}$ (modified eddy viscosity) and can be prescribed as $\bar{\nu} = 0.033\,\bar{i}\,\bar{v}\,\bar{L}$.

This condition is usually applied to inlet boundaries.

5.4.2 Prescribed turbulence K-ω

prescribed turbulence K-ω

This condition can only be applied to a K-ω model. Time-dependent turbulence variables are directly prescribed

$$K = \bar{K}(t)$$
$$\omega = \bar{\omega}(t)$$

and applied to boundaries. The K and ω equations at boundary nodes are then replaced by these conditions.

In addition to directly given values, they can also be prescribed indirectly via

$$K = \tfrac{3}{2}\left(\bar{i}\,\bar{v}\right)^2$$
$$\omega = K^{\frac{1}{2}} \Big/ \left(0.27\bar{L}\right)$$

where, $\bar{v}, \bar{L}$ and $\bar{i}$ $(= 0.01 \sim 0.1)$ are velocity scale, length scale and turbulence intensity respectively.

This condition is usually applied to inlet boundaries.

5.4.3 Zero flux of turbulence variables

The natural boundary condition of any equation requires a specified normal flux of that variable. The turbulence flux condition is currently unavailable in ADINA-F (practically such conditions are not available). Therefore a zero flux is implicitly assumed along the boundary where no turbulence condition is explicitly specified.

5.4.4 Solid walls

This condition is probably the most important one in modeling turbulence. Let us take the K-ε model as an example. At a solid wall, the no-slip condition applies so that both the mean and fluctuating velocities are zero. In the near wall region where $0< y^+ <100$, the wall function is applied. In this region, the Reynolds stresses $\overline{\rho \mathbf{v}'\mathbf{v}'}$ are nearly constant, so is the kinetic energy K. The convection and diffusion in the K-equation are negligible so that local equilibrium prevails. When buoyancy effects are absent, this implies that all source terms in the K-equation are zero. We further recall the definition of the shear stress and obtain

$$\mu_t (v')^2 = \rho\varepsilon, \quad \mu_t v' = \rho v_*^2$$

The solutions of the above equations determine the variables K and ε in the near-wall region

$$K_w = \frac{v_*^2}{\sqrt{c_\mu}}, \quad \varepsilon_w = \frac{\rho v_*^4}{\mu_0} W'(y_w^+)$$

Similarly, the boundary values of turbulence variables in K-ω models are obtained

$$K_w = \frac{v_*^2}{\sqrt{\alpha\beta_K}}, \quad \omega_w = \frac{\rho v_*^2}{\mu_0}\sqrt{\frac{\alpha}{\beta_K}} W'(y_w^+)$$

The condition for momentum equations is the friction force computed by integrating the shear stress ρv_*^2

$$\mathbf{F}_\tau = -\oint h_i \rho v_*^2 \boldsymbol{\tau} dS$$

and added to the right hand side of the momentum equations, where $\boldsymbol{\tau}$ is the tangential direction of the wall.

Wall temperature can be specified in two ways. The first one is to directly apply usual or special thermal conditions as done in laminar models. In this method, temperature condition must not be specified in wall boundary condition. The second method is to specify a thermal condition in the wall boundary condition, such that the wall function shall be incorporated into the temperature condition. This method is described below.

Reynolds's analogy between momentum and energy transport gives a similar law for mean temperature. The law-of-the-wall for temperature implemented in ADINA-F is

$$T^+\left(y^+, \mathrm{Pr}, \sigma_\theta\right) = \begin{cases} \mathrm{Pr}\, y^+ & y^+ < y_\theta^+ \\ \sigma_\theta W\left(y^+\right) + \sigma_\theta P_\theta & y^+ \geq y_\theta^+ \end{cases} \qquad (1.2)$$

where P_θ is computed by using the formula given by Jayatilleke

$$P_\theta = 9.24\left[\left(\mathrm{Pr}/\sigma_\theta\right)^{\frac{3}{4}} - 1\right]\left(1 + 0.28e^{-0.007\mathrm{Pr}/\sigma_\theta}\right)$$

Given the molecular Prandtl number, the thermal sublayer thickness y_θ^+ is computed as the y^+ value at which the two curves in Eq. (1.2) intersect.

The formula

$$T^+ = \frac{\rho v_* C_v \left(\theta_w - \theta\right)}{q_w}$$

is then used to determine the wall heat flux or wall temperature, depending on the type of the thermal wall boundary condition (specified temperature $\overline{\theta}_w$ or heat flux $\overline{q}_w$).

The near wall treatment of the fluid and the turbulence variables, as discussed above, is automatically enforced whenever a wall condition or its analogue is applied. These conditions are

- No-slip fixed/moving wall conditions
- Moving wall condition (type=tangential)
- Moving wall condition (type=rotational)
- No-slip fluid-structure interface
- Fluid-structure interface (type=tangential)
- Fluid-structure interface (type=rotational)

5.4.5 User-supplied boundary condition

These conditions are defined in the same user-supplied subroutines and applied in the same manner as for fluid flows and temperature conditions. The detailed description is given in Section 3.4.

5.5 Initial conditions

In transient analyses, all solution variables ($\mathbf{v}, p, \theta, K, \varepsilon$) or ($\mathbf{v}, p, \theta, K, \omega$) must be specified, depending on the turbulence model selected. The default initial conditions to all variables are zero conditions. Although initial conditions are not required in steady-state analyses, they are used as a "guessed" solution in the equilibrium iterations. A good initial condition, as will be seen in Chapter 13, may accelerate the convergence during equilibrium iterations. In certain cases, the initial condition may become a key factor in obtaining converged solutions.

In low-speed compressible flows, recalling that pressure, temperature and density must satisfy the state equation of the ideal gas law, zero pressure or zero temperature must not be specified. Since the default initial conditions to all variables are zero, actual initial conditions are always required in this case in both transient and steady-state analyses.

In turbulent flows, initial conditions can be selected as the same as the value of turbulent variables at inlet boundary conditions.

Initial conditions can be improved in restart analyses or using the solutions mapped from a similar solution.

5.6 Material models

5.6.1 K-ε turbulence model

material model: K-ε turbulence

This material model can only be applied to the turbulence K-ε model. In this model, both the fluid properties and the empirically determined model constants must be input.

The fluid properties are

ρ = fluid density
μ = fluid viscosity
$\mathbf{g}$ = gravitational acceleration vector

C_p = specific heat at constant pressure
C_v = specific heat at constant volume
k = thermal conductivity
q^B = rate of heat generated per unit volume
β = thermal expansion coefficient
θ_0 = reference temperature in buoyancy force
σ = coefficient of surface tension
κ = bulk modulus of elasticity

As mentioned in Section 3.6, not all these parameters are required in all problems. The parameters that must be input depend on the problem to be solved.

In incompressible flows considered here, the bulk modulus of elasticity κ and the specific heat at constant volume C_v are not required. In this case, κ is assumed to be infinity and C_v is forced to be equal C_p. If heat transfer is not considered, C_p, k, q^B, β, and θ_0 can be ignored.

In slightly compressible flows, C_v is forced to be equal to C_p and, therefore, can be ignored. C_p, k, q^B, β and θ_0 can also be omitted if heat transfer is not included.

In low-speed compressible flows, κ is not used in the computations and therefore ignored. The density is also predetermined through the state equation and thus not required. All other material data must be input.

The above convention is used throughout the chapter.

Besides the above fluid properties, the additional model constants c_μ, c_1, c_2, c_3, σ_K, σ_ε, σ_θ, d_w^+ and κ_0 must also be specified. Their default values are

$$c_\mu = 0.09, \quad c_1 = 1.44, \quad c_2 = 1.92,$$
$$c_3 = 0.8, \quad \sigma_K = 1, \quad \sigma_\varepsilon = 1.3,$$
$$\sigma_\theta = 0.9, \quad d_w^+ = 70, \quad \kappa_0 = 0.4$$

5.6.2 RNG K-ε turbulence model

material model:RNG K-ε turbulence

This material model can only be applied to the turbulence K-ε model. The data required in this model is the same as that in the K-ε turbulence K-ε model. The default values of the empirical constants are the same as for the standard K-ε model. Another conventional set of these data are

$$\begin{aligned} c_\mu &= 0.085, & c_1 &= 1.42, & c_2 &= 1.68, \\ c_3 &= 0.8, & \sigma_k &= 0.7179, & \sigma_\varepsilon &= 0.7179, \\ \sigma_\theta &= 0.9, & d_w^+ &= 70, & \kappa_0 &= 0.4 \end{aligned}$$

5.6.3 High Reynolds K-ω turbulence model

material model: K-ω high-Reynolds turbulence

This material model can only be applied to the turbulence K-ω model. Besides the required fluid properties as defined in the K-ε material model**,** the additional model constants for turbulence variables, α, α_ω, β_K, β_ω, σ_K, σ_ω, σ_θ, β_θ, d_w^+ and κ_0 must also be specified. Their default values are

$$\begin{aligned} \alpha &= 1, & \alpha_\omega &= 5/9, & \beta_K &= 0.09, \\ \beta_\omega &= 0.075, & \sigma_K &= 2, & \sigma_\omega &= 2, \\ \beta_\theta &= 0.712, & \sigma_\theta &= 1, & d_w^+ &= 70, \\ \kappa_0 &= 0.41 \end{aligned}$$

5.6.4 Low Reynolds K-ω turbulence model

material model: K-ω low-Reynolds turbulence

This material model contains the same input as the high Reynolds K-ω model and can only be applied to the turbulence K-ω model.

5.6.5 Shear Stress Transport model

This material model can only be applied to the K-ω turbulence model. The constants for turbulence variables are set automatically and their values are:

$$\begin{array}{lll} \sigma_{K_1} = 1.176, & \sigma_{K_2} = 1.0, & \sigma_{\omega_1} = 2.0, \\ \sigma_{\omega_2} = 1.168, & \alpha = 1.0, & \alpha_1 = 0.553, \\ \alpha_2 = 0.553, & \beta_1 = 0.075, & \beta_2 = 0.0828, \\ a_1 = 0.31 & d_w^+ = 70 & \kappa_0 = 0.41 \end{array}$$

The Low-Reynolds and High Reynolds variants are available in the SST material model. In addition, the model is available for FCBI and FCBI-C elements.

5.6.6 Spalart-Allmaras and Detached Eddy Simulation material models

This material model can only applied to SA and DES turbulence models. Besides the fluid properties, there are additional constants that must be specified. These constants and their default values are:

$$\begin{array}{lll} c_{w2} = 0.3, & c_{w3} = 2.0, & \kappa = 0.41, \\ c_{b1} = 0.1355, & c_{b2} = 0.622, & c_{v1} = 7.1, \\ \sigma_s = 2/3, & C_{prod} = 2, & \sigma_\theta = 1, \\ C_{DES} = 6.4, & c_{w1} = c_{b1}/\kappa^2 + (1 + c_{b2})/\sigma_s & \end{array}$$

The SA and DES models are available for FCBI and FCBI-C elements. In both pre- and post-processing the turbulent kinetic energy K (as in the K-ε model), through all input and output, represents $\tilde{\nu}$. Therefore, when prescribing $\tilde{\nu}$, specifying its initial value, relaxation factors, and other input parameters, K as displayed on AUI and in its commands becomes $\tilde{\nu}$.

5.6.7 Two-layer zonal material model

This model is always associated with K-ε model. Its material includes all data for K-ε model plus some additional ones, names $A_\mu, A_\varepsilon, c_l, R_{c\min}, R_{c\max}, L_w$. Their default values are determined by the program and explained in Chapter 2.

5.6.8 Material curves

See corresponding section in Chapter 3 for details.

5.6.9 Gas models

See corresponding section in Chapter 3 for details.

5.6.10 K-ε realizable turbulence model

This material model has the same input as the standard K-ε turbulence model.

This page intentionally left blank

Chapter 6 Formulation of high-speed compressible flows

In this chapter, we introduce our formulations for high-speed compressible flows. This model can be coupled with ADINA solid models and be associated with mass transfer solutions.

ref. Bathe, K.J., Zhang, H., and Ji, S., "Finite Element Analysis of Fluid Flows fully Coupled with Structural Interactions," *Computers & Structures*, Vol.72, pp.1-16, 1999.

ref. Bathe, K.J., Zhang, H., and Zhang, X., "Some Advances in the Analysis of fluid flows," *Computers & Structures*, Vol.64, pp.909-930, 1997.

ref. Bathe, K.J., Zhang, H., and Wang, M.H., "Finite Element Analysis of Incompressible and Compressible fluid flows with Free Surfaces and Structural Interactions," *Computers & Structures*, Vol.56, pp.193-214, 1995.

ref. Bathe, K.J., Walczak, J., and Zhang, H., "Some Recent Advances for Practical Finite Element Analysis," *Computers & Structures*, Vol.47, pp.511-521, 1993.

The following table presents a quick overview of the capabilities that are available for high-speed compressible flows.

Table 6-1 Functionalities in the high-speed compressible flow model

Category	**Functionality**
Coupled models Note: Heat transfer must be coupled.	ADINA solid model
	solid element groups
	mass transfer
Computational domains	2D planar, 2D axisymmetric and 3D

Analyses	steady-state and transient	
Elements	3-node triangle	
	4-node tetrahedron	
Material models	Constant	
	temperature-dependent	
	pressure-dependent	
	pressure-temperature-dependent	
	user-supplied	
Boundary conditions Note: The column furthest to the right represents the solution variables or equations that are immediately affected by the specified condition. The solution variable **d** indicates a moving boundary condition.	prescribed specific discharges	$\rho\mathbf{v}$
	zero specific discharges	
	prescribed density	
	concentrated force load	
	distributed normal-traction load	
	field centrifugal load	
	prescribed total energy	ρE
	concentrated heat flow load	
	distributed heat flux load	
	fixed wall	$\mathbf{v}, \theta$
	angular velocity	
	moving wall	$\mathbf{v}, \theta, \mathbf{d}$
	external flow	$\rho, \rho\mathbf{v}, \rho E$
	supersonic at inlet	
	subsonic at inlet	
	supersonic at outlet	
	subsonic at outlet	
	symmetric	
	fluid-structure interface	$\mathbf{v}, \theta, \mathbf{d}$
	Gap	all
	User-supplied	
Initial conditions Note: Zero pressure or temperature cannot be specified.	Zero (default conditions)	
	specified	
	mapped from other solutions	
	restart run	
Solvers for linearized equations	Gauss elimination method (COLSOL)	
	sparse solver	
	iterative methods (RPBCG, RPGMRES, and multi-grid)	
Other capabilities	automatically nondimensional procedure	
	automatic time-stepping CFL option	
	automatic time-stepping ATS option	
	skew system	
	constraint condition	
	conjugate heat transfer	
	pressure datum	

6.1 Governing equations

The governing equations used in our formulations are the conservative forms of the Navier-Stokes equations that are listed in Table 2-1. For moving mesh problems, the ALE formulation described in Section 2.14 is used. The solution variables are the conservative variables $(\rho, \rho\mathbf{v}, \rho E)$ (although the primitive variables are still used in the input phase for the purpose of convenience).

6.2 Numerical method

The numerical method used here in solving the governing equations combines a finite volume approach to the inviscid terms and a finite element approach to the viscous terms.

The basic step of a finite volume procedure is the calculation of the flux at the interface of a control volume. The total flux consists of the inviscid Eulerian flux term, $\mathbf{F}_n$, that includes the convective term and pressure, and the viscous term, $\mathbf{G}_n$, that consists of viscous and conductive terms. We calculate the flux $\mathbf{F}_n$ using a finite volume-based flux-splitting method while we calculate the viscous term $\mathbf{G}_n$ using a finite element-based method.

In two-dimensional geometries, the control volume is based on the use of the triangular element. The faces of a control volume consist of straight lines. In three-dimensional geometries, the control volume is based on the uses of the tetrahedral element. The faces of a control volume consist of rectangular planes and each of the rectangular consists of two triangles. Each control volume is divided into a few sub-control volumes that are located in the neighboring elements. They are detailed in Section 6.3.

Let us integrate the governing equation, in an exact manner on a general moving control volume V enclosed by its faces S

$$^{t+\alpha\Delta t}(V\mathbf{U}) - {}^{t}(V\mathbf{U}) + \int_{t}^{t+\alpha\Delta t}\left[\oint (\mathbf{F}_n + \mathbf{G}_n)dS - \int_V \mathbf{C}dV\right]dt = \mathbf{0} \quad (1.3)$$

The surface and volume integrations are then approximated by some averaged values denoted by double-over bars

$$^{t+\alpha\Delta t}(V\mathbf{U}) - {}^{t}(V\mathbf{U}) + \alpha\Delta t\left[\sum(\bar{\bar{\mathbf{F}}}_n + \bar{\bar{\mathbf{G}}}_n)S - \bar{\bar{\mathbf{C}}}V\right] = \mathbf{0}$$

The above terms without superscripts are calculated at time $t + \alpha\Delta t$ if an implicit method is used, or at time t if an explicit method used. We will introduce our approaches to these averaged terms in the following sections.

6.2.1 Geometric conservation law in moving meshes

ref. Zhang, H., Reggio, M., Trépanier, J.Y., and Camarero, R., "Discrete Form of the GCL for Moving Meshes and Its Implementation in CFD schemes," *Computers Fluids*, Vol.22, No.1, pp.9-23, 1993.

Consider a special flow pattern: stationary flow with constant density, pressure and temperature. In this trivial case, Eq.(1.3) becomes

$$^{t+\alpha\Delta t}V - {}^{t}V = \sum\int_{t}^{t+\alpha\Delta t}\int_{S} w_n dS dt \equiv \sum\Delta V \qquad (1.4)$$

This equation states that during a time interval the increased volume equals the volumetric increase as a consequence of the motion of the faces. Eq.(1.4) is not an additional equation that needs to be solved, but must be satisfied implicitly as a fundamental geometric rule. It is so called the geometric conservation law (GCL). Any violation of GCL will produce errors to the flow field calculation, resulting in numerical oscillations or unsuccessful solution procedure.

The key point here is to define some averaged mesh velocity $\bar{w}_n$ on the faces and embed it into the numerical scheme such that the GCL is exactly satisfied. Our task here is to evaluate the volumetric increments in an exact manner in different flow fields. In order to simplify the notation, we first consider the volumetric increments in a time step Δt.

First we define the nodal mesh velocity as

$$\mathbf{w}_i = \frac{{}^{t+\Delta t}\mathbf{r}_i - {}^{t}\mathbf{r}_i}{\Delta t}$$

Thus, the moving nodal coordinates can be expressed as

$$\mathbf{r}_i = {}^{t}\mathbf{r}_i + \lambda \Delta t\, \mathbf{w}_i, \qquad \lambda \in [0,1]$$

In two-dimensional cases, two points define a face, say 1 and 2, and we use the local coordinate to represent the line

$$\mathbf{w} = \mathbf{w}_1 + \xi(\mathbf{w}_2 - \mathbf{w}_1)$$
$$\mathbf{r} = \mathbf{r}_1 + \xi(\mathbf{r}_2 - \mathbf{r}_1), \qquad \xi \in [0,1]$$

Using these transformations we can obtain the volumetric increment

$$\Delta V(\Delta t) = \Delta t \int_0^1 \int_0^1 \mathbf{w} \bullet \mathbf{S}\, d\xi d\lambda = \Delta t \mathbf{w}_0 \bullet {}^{t+\Delta t/2}\mathbf{S}$$

where $\mathbf{w}_0 = (\mathbf{w}_1 + \mathbf{w}_2)/2$.

In the axisymmetric case, similarly, we can obtain

$$\begin{aligned}\Delta V(\Delta t) &= \Delta t \int_0^1 \int_0^1 y \mathbf{w} \bullet \mathbf{S}\, d\xi d\lambda \\ &= \frac{\Delta t}{6}(2\,{}^{t+\Delta t/2}y_1 + {}^{t+\Delta t/2}y_2)\mathbf{w}_1 \bullet {}^{t}\mathbf{S} \\ &\quad + \frac{\Delta t}{6}(2\,{}^{t+\Delta t/2}y_2 + {}^{t+\Delta t/2}y_1)\mathbf{w}_2 \bullet {}^{t}\mathbf{S} \\ &\quad + \frac{\Delta t^2}{4}({}^{t+2\Delta t/3}y_1 + {}^{t+2\Delta t/3}y_2)\mathbf{w}_1 \times \mathbf{w}_2\end{aligned}$$

In the three-dimensional case, we only need to consider triangular surfaces. Let points 1, 2 and 3 define the vertexes of the triangle, the local transformations are

$$\mathbf{w} = \mathbf{w}_1 + \xi(\mathbf{w}_2 - \mathbf{w}_1) + \eta(\mathbf{w}_3 - \mathbf{w}_1)$$
$$\mathbf{r} = \mathbf{r}_1 + \xi(\mathbf{r}_2 - \mathbf{r}_1) + \eta(\mathbf{r}_3 - \mathbf{r}_1), \qquad \xi,\ \eta,\ 1-\xi-\eta \in [0,1]$$

We can then obtain

$$\Delta V(\Delta t) = \Delta t \int_0^1 \int_0^1 \int_0^{1-\eta} \mathbf{w} \bullet \mathbf{S} d\xi d\eta d\lambda$$
$$= \Delta t \mathbf{w}_0 \bullet {}^{t+\Delta t/2}\mathbf{S} + \frac{\Delta t^3}{24}(\mathbf{w}_1 \times \mathbf{w}_2) \bullet \mathbf{w}_3$$

where $\mathbf{w}_0 = (\mathbf{w}_1 + \mathbf{w}_2 + \mathbf{w}_3)/3$. Note that the volumetric increment of a rectangular surface is the sum of the volumetric increments of the two triangles that form the rectangular.

Using the computed volumetric increment, we can then define our averaged mesh velocity from time t to time $t + \alpha \Delta t$ on the interface

$$\hat{w}_n = \frac{\Delta V(\alpha \Delta t)}{\alpha \Delta t \, {}^{t+\alpha \Delta t}S}$$

This averaged mesh velocity computed on each face will be used everywhere on that face regardless of how other variables are computed. The GCL is therefore always satisfied.

6.2.2 Flux-Splitting method

The inviscid flux term $\mathbf{F}_n$ deserves special attention, not only because it dominates the total flux in the outflow region, but also because wave motions and discontinuities in the flow field are caused by this term. When the viscous term is negligible, the governing equations form a hyperbolic system.

Consider the following one-dimensional hyperbolic equation

$$\frac{\partial U}{\partial t} = \frac{\partial f(U)}{\partial x} \approx \frac{f_{i+1/2} - f_{i-1/2}}{\Delta x}$$

The problem is to evaluate the numerical fluxes f at the positions $i + 1/2$ and $i - 1/2$. An upwinding scheme can be constructed as

$$f_{i+1/2} = \frac{1}{2}\left[(f_i + f_{i+1}) - \left| f'_{i+1/2} \right| (U_{i+1} - U_i) \right] \qquad (1.5)$$

If the wave speed, represented here by f', is properly evaluated such that

$$f_{i+1} - f_i \equiv f'_{i+1/2}(U_{i+1} - U_i) \qquad (1.6)$$

Eq.(1.5) becomes

$$f_{i+1/2} = \begin{cases} f_i & \text{if } f'_{i+1/2} > 0 \\ f_{i+1} & \text{if } f'_{i+1/2} < 0 \end{cases}$$

This gives the solution of the so-called Riemann problem and a program using the above scheme is called a Riemann solver.

For a full Eulerian equation system, approximate solutions of the Riemann scheme are widely used. The method used here is based on Roe's scheme. In this scheme, a set of averaged variables is used to approximate the Jacobian matrix $\mathbf{A}(\mathbf{U}) = \partial \mathbf{F}_n(\mathbf{U}) / \partial \mathbf{U}$, such that Eq.(1.6) is exactly satisfied.

Considering any face shared by two control volumes, denoted as states *L* and *R* that refer to the left and right volumes respectively, the Roe average variables are given by

$$\hat{\rho} = \sqrt{\rho_L \rho_R}$$

$$\hat{\mathbf{\Phi}} = \frac{\sqrt{\rho_L}\mathbf{\Phi}_L + \sqrt{\rho_R}\mathbf{\Phi}_R}{\sqrt{\rho_L} + \sqrt{\rho_R}}, \qquad \mathbf{\Phi} = (\mathbf{v}, H)$$

Other variables are computed as functions of the Roe averaged variables. For example, the conservative variables and flux increments are computed by

$$\hat{\mathbf{U}} = \begin{bmatrix} \hat{\rho} \\ \hat{\rho}\,\hat{\mathbf{v}} \\ \left(\frac{1}{2}\alpha \hat{\mathbf{v}} \bullet \hat{\mathbf{v}} + \hat{H} \right) \hat{\rho} / \gamma \end{bmatrix}, \quad \Delta \mathbf{F} = \hat{\mathbf{A}} \Delta \mathbf{U} = \hat{\mathbf{P}} \hat{\mathbf{D}} \hat{\mathbf{P}}^{-1} \Delta \mathbf{U}$$

where

$$\Delta\mathbf{U} = \mathbf{U}_R - \mathbf{U}_L$$
$$\Delta\mathbf{F} = \mathbf{F}_n(u_R - \hat{w}_n, \mathbf{U}_R) - \mathbf{F}_n(u_L - \hat{w}_n, \mathbf{U}_L) \equiv \mathbf{F}_{nR} - \mathbf{F}_{nL}$$
$$\hat{f} = f(\hat{\mathbf{U}}), \qquad f = \mathbf{A}, \mathbf{P}, \mathbf{D} \text{ and } \mathbf{P}^{-1}$$

It is important to note that the mesh velocity is always computed according to GCL along the interface. The averaged inviscid flux on an interface is therefore computed by

$$\bar{\bar{\mathbf{F}}}_n = \frac{1}{2}\left[\mathbf{F}_{nL} + \mathbf{F}_{nR}\right] - \frac{1}{2}\hat{\mathbf{P}}\left|\hat{\mathbf{D}}\right|\hat{\mathbf{P}}^{-1}\Delta\mathbf{U} \tag{1.7}$$

where

$$\left|\mathbf{D}\right| = \text{diag}\left\{\left|\Delta u + c\right|, \left|\Delta u - c\right|, \left|\Delta u\right|, \left|\Delta u\right|, \left|\Delta u\right|\right\}$$
$$\Delta u = u - w_n$$

6.2.3 Finite element method

We use the finite element method to compute the viscous terms. For both the triangular element and the tetrahedral element, the viscous terms can be calculated exactly along the control volume faces. Thus we have

$$\int_S \mathbf{G}_n dS \equiv \bar{\bar{\mathbf{G}}}_n S$$

For example, using the element shape functions h_i, the heat flux is computed as

$$\int_S \mathbf{q} \bullet d\mathbf{S} = \int_S -k\,\mathbf{n} \bullet \nabla h_i \theta_i dS = -S\,k\,\mathbf{n} \bullet \nabla\, h_i \theta_i$$

Here, of course, the temperature must be evaluated using conservative variables while the global matrix system is assembled. The interpolation functions and their computations are detailed in Chapter 10.

6.2.4 Solution procedure

The procedure that we have developed for the calculation of high-speed compressible flows is basically the same as that for incompressible flows when Newton-Raphson method is used. In forming the linear equations for high-speed compressible, the following steps are performed, where N_e is the number of elements, and n_f and n_c are the numbers of sub-volumes and sub-faces, respectively, within the element,

$$
\begin{array}{l}
\textit{for } i_e = 1,2,...,N_e \\
\left\{ \begin{array}{l}
V, \nabla h_i = ... \\
\textit{for } i_f = 1,2,...,n_f \\
\quad \left\{ \begin{array}{l}
S, \hat{w}_n = ... \\
f = S(\bar{\bar{\mathbf{F}}}_n + \bar{\bar{\mathbf{G}}}_n),\ \partial f / \partial \mathbf{U} = ... \\
\quad \left(f, \partial f / \partial \mathbf{U}\right) \rightarrow \textit{left control volume} \\
-\left(f, \partial f / \partial \mathbf{U}\right) \rightarrow \textit{right control volume}
\end{array} \right. \\
\textit{for } i_c = 1,2,...,n_c \\
\quad \left\{ \left(\alpha \Delta t\right)^{-1} \left[{}^{t+\alpha\Delta t}(V\mathbf{U}) - {}^{t}(V\mathbf{U}) \right] - V\bar{\bar{\mathbf{C}}} \rightarrow ... \right\}
\end{array} \right.
\end{array}
$$

6.2.5 Explicit time integration scheme

Explicit time integration of the method described in the previous sections is available for high-speed compressible flows. However, it cannot be used with any moving boundary conditions.

The time step size is restricted by the well-known Courant-Friedrichs-Lewy (CFL) stability condition

$$\Delta t \le \min\left\{\frac{\Delta x}{|\Delta u| + c + 2\mu/\rho\Delta x}\right\} \equiv \Delta t_c$$

where Δx is the element size. The explicit method should always be associated with the CFL automatic time stepping option (see Chapter 12 for details), which allows the program to determine the sub-time-step size for a specified factor λ_{CFL}

$$\Delta t_i = \lambda_{\text{CFL}} \Delta t_c$$

In order to keep the scheme stable, the factor λ_{CFL} must be smaller than 1, usually about 0.8.

The explicit procedure is summarized here. We start with the initial guess of the solution **U**, which is the solution at time t. In the first time step, the initial condition is used. The following procedure is used to obtain the solution at $t + \Delta t$: Start with the current solution time $t_i = t$,

$$\begin{array}{l}
\textit{until}\ (t_i \equiv t + \Delta t) \\
\left\{\begin{array}{l}
\mathbf{U}^0 = {}^{t_i}\mathbf{U},\ V^0 = {}^{t_i}V\ \textit{and compute}\ \Delta t_i \\
\textit{for}\ i_e = 1,2,...,N_e \\
\left\{\begin{array}{l}
V, \nabla h_i = ... \\
\textit{for}\ i_f = 1,2,...,n_f \\
\left\{\begin{array}{l}
S, \hat{w}_n = ...,\ f = S(\bar{\bar{\mathbf{F}}}_n + \bar{\bar{\mathbf{G}}}_n) \\
(f, -f) \rightarrow (\textit{left}, \textit{right})\ \textit{control volumes}
\end{array}\right\} \\
\textit{for}\ i_c = 1,2,...,n_c \\
\left\{\mathbf{U} = V^{-1}\left[V^0\mathbf{U}^0 - \Delta t_i\left(\sum(\bar{\bar{\mathbf{F}}}_n + \bar{\bar{\mathbf{G}}}_n)S - \bar{\bar{\mathbf{C}}}V\right)\right]\right\}
\end{array}\right\} \\
t_i = t_i + \Delta t_i,\ {}^{t_i}\mathbf{U} = \mathbf{U},\ {}^{t_i}V = V
\end{array}\right\}
\end{array}$$

6.3 Control volumes and finite elements

6.3.1 Control volume in triangular elements (3-node)

A control volume for two-dimensional planar and axisymmetric flows is obtained based on the triangular elements as defined in the figure below. The 3-node triangular element is detailed in Chapter 10.

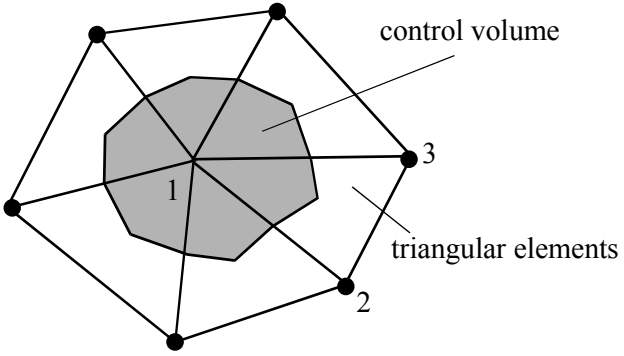


Figure 6.1 Definition of a two-dimensional control volume

Each triangular element is divided into three sub-control volumes with nodes 1, 2 and 3 respectively, by connecting the center point to the midpoints of the sides. These are three internal faces that are shared by the three pairs of sub-control volumes. The six other element faces, every two of them forming a side of the triangle, may be internal faces shared with another part of the same control volume, or physical boundary sides that are on the boundary of the computational domain.

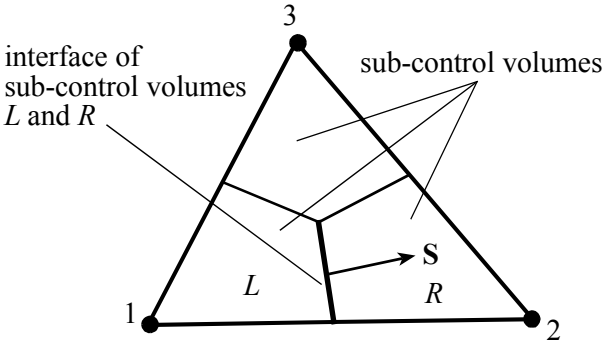


Figure 6.2 Sub-control volumes and their interfaces within one triangular element

All conservative variables are defined at the corner nodes, representing averaged quantities over control volumes. A straightforward computation of flux terms is carried out on the internal faces and assembled in the two control volumes that share the face.

On a physical boundary face, the flux term will either be computed in conjunction with the applied boundary condition, or the boundary condition is used to replace the corresponding equation defined at that node, depending on the type of condition.

Non-physical boundary faces are skipped in the computation loop since the flux is cancelled out.

In computing viscous terms, the coordinates and all primitive variables are interpolated by

$$f = \sum_{i=1}^{3} h_i f_i \qquad f = \mathbf{x}, \mathbf{v}, p \text{ and } \theta$$

where, the interpolation functions are the same as used for incompressible flows and defined in Chapter 10.

6.3.2 Control volume in tetrahedral elements (4-node)

A similar control volume for three-dimensional flows is obtained based on tetrahedral elements as defined in the figure below. The 4-node tetrahedral element is detailed in Chapter 10.

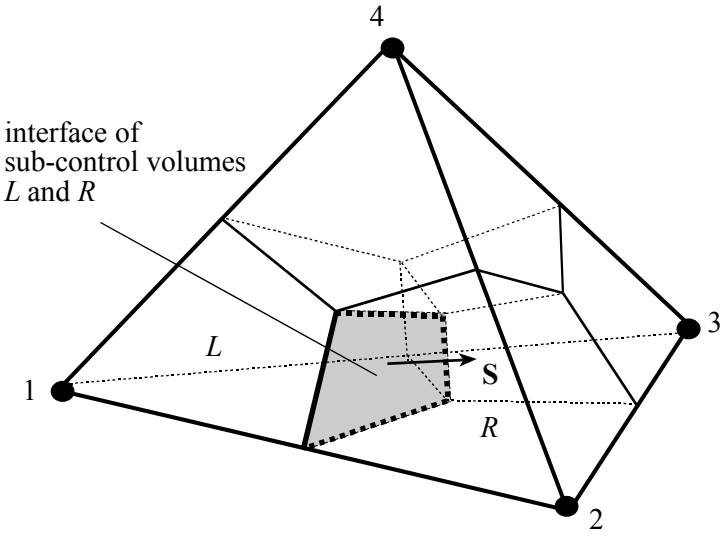


Figure 6.3 Sub-control volumes and their interfaces within one tetrahedral element

Each tetrahedron is divided into 4 sub-control volumes containing the nodes 1, 2, 3 and 4, respectively, by connecting the center point of the tetrahedron and the center points of the four triangular surfaces. The triangular surfaces are subdivided by connecting their center points to their corresponding midpoints of the sides. There are a total of six internal faces shared by the six pairs of sub-control volumes. The twelve other element faces, every three of them forming a triangular surface of the tetrahedron, may be internal faces that are shared with another part of the same control volume, or physical boundary faces that are part of the boundary of the computational domain.

All conservative variables are defined at corner nodes, representing averaged quantities over control volumes. Similar to the two-dimensional

case, the computation of flux terms is only carried out on the internal faces and the physical boundary faces.

Regarding the computation of viscous terms, the coordinates and all primitive variables are interpolated using

$$f = \sum_{i=1}^{4} h_i f_i \qquad f = \mathbf{x}, \mathbf{v}, p \text{ and } \theta$$

where, the interpolation functions are the same as used for incompressible flows and defined in Chapter 10.

6.4 Boundary conditions

6.4.1 General descriptions

Using proper boundary conditions is an essential step towards obtaining the solution of the fluid flow problem. The boundary conditions strongly depend on the flow types.

In high-speed compressible flows, the velocity, pressure and temperature are strongly coupled and must be treated like one variable as far as the boundary conditions are concerned.

Generally, the unsteady viscous equations for compressible flows form a hybrid parabolic-hyperbolic system. Problems to be solved based upon such a system are boundary-value problems. Only a few special mathematical results are known to ensure the existence and uniqueness of the solution. This is contrary to the incompressible flow case for which the mathematical theory is well established.

On the other hand, the Euler equations consist of a hyperbolic system, for which the theory about boundary conditions has been established for decades. The special boundary conditions developed here are mainly based on the theory for inviscid flows. Practically speaking, a boundary condition set that is suitable for inviscid flows may be a good first consideration for viscid flows as well. We will use the following guidelines in the discussion of our special boundary conditions. Let the inviscid flux (Eq.(1.7)) on the

boundary be equal to that in the right control volume (representing the boundary "volume")

$$\mathbf{F}_{nR} = \frac{1}{2}\left[\mathbf{F}_{nL} + \mathbf{F}_{nR}\right] - \frac{1}{2}\hat{\mathbf{P}}\left|\hat{\mathbf{D}}\right|\hat{\mathbf{P}}^{-1}\Delta\mathbf{U} \qquad (1.8)$$

Defining

$$\begin{bmatrix}\eta_1\\ \eta_2\\ \eta_3\\ \eta_4\\ \eta_5\end{bmatrix} = \hat{\mathbf{P}}^{-1}\Delta\mathbf{U} = \begin{bmatrix}\left(\Delta p + \hat{\rho}\hat{c}\Delta u\right)/(2\hat{c}^2)\\ \left(\Delta p - \hat{\rho}\hat{c}\Delta u\right)/(2\hat{c}^2)\\ \Delta\rho - \Delta p/\hat{c}^2\\ \Delta(\mathbf{v}\bullet\boldsymbol{\tau}_1)\\ \Delta(\mathbf{v}\bullet\boldsymbol{\tau}_2)\end{bmatrix}$$

we have the characteristic forms of Eq.(1.8)

$$(\hat{d}_i + \left|\hat{d}_i\right|)\eta_i = 0, \qquad i = 1,2,3,4,5 \qquad (1.9)$$

where d_i are the eigenvalues defined in **D,** namely

$$d_1 = \Delta u + c, \quad d_2 = \Delta u - c, \quad d_i = \Delta u \ \ (i = 3,4,5)$$

Eq.(1.9) states that the characteristic equation η_i must be zero if its corresponding eigenvalue d_i is positive. The condition $\eta_4 = \eta_5 = 0$ is trivial, which corresponds to the zero derivative of the tangential velocity with respect to the normal direction. d_1, d_2 and d_3 are the three wave speeds along the characteristic lines. According to the signs of these wave speeds, flows are classified into different regimes, namely supersonic at inlet, subsonic at inlet, subsonic at outlet and supersonic at outlet, as illustrated in the following figure. Boundary conditions strongly depend on the status of flow regimes on the boundaries.

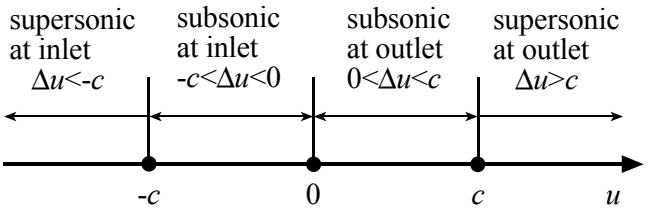


Figure 6.4 Definition of flow regimes in high-speed compressible flows

The most frequently encountered boundary is probably the solid wall boundary, which can be fixed, with prescribed motions or deformable structures. Along a slip-wall, which is usually used in Euler equations, the normal velocity is specified as the speed of the solid, while the tangential velocities are left free to be determined by the solution. Along a no-slip wall, all velocity components are prescribed to be equal to the velocity of the solid. Thermal conditions on walls are usually either specified temperature or adiabatic conditions. Since the relative velocity Δu is zero, only the first wave speed is positive and, therefore, the additional condition is provided by $\eta_1 = 0$.

Besides walls, other boundaries are exposed to the environment and called open boundaries. A flow is called external if a part of the open boundary is "infinite". Otherwise it is called internal. Typical examples of external flows are aerodynamic flows around aircrafts, space shuttles, missiles, etc. Internal flows are widely visible in instruments that deal with air.

In internal flows, an open boundary can be an inlet or an outlet if the outward velocity is negative or positive, respectively. When a flow is supersonic at an inlet, all wave speeds are negative, which corresponds to incoming waves from outside of the computational domain. The boundary values of the fluid variables are then completely determined by these waves. Therefore three values must be specified on the boundary (see the figure below).

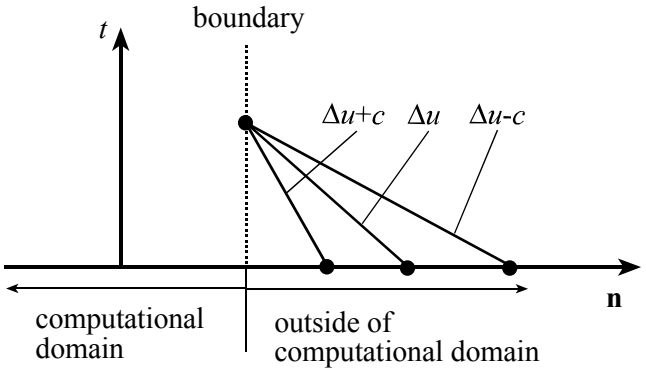


Figure 6.5 Illustration of the supersonic condition at inlet

When a flow becomes subsonic at an inlet, two wave speeds are negative while one (d_1) is positive. The negative wave speeds correspond to two incoming waves from outside the computational domain while the positive speed indicates that one wave leaves the domain. Therefore two values must be specified on the boundary (see the figure below). The additional condition is given by $\eta_1 = 0$.

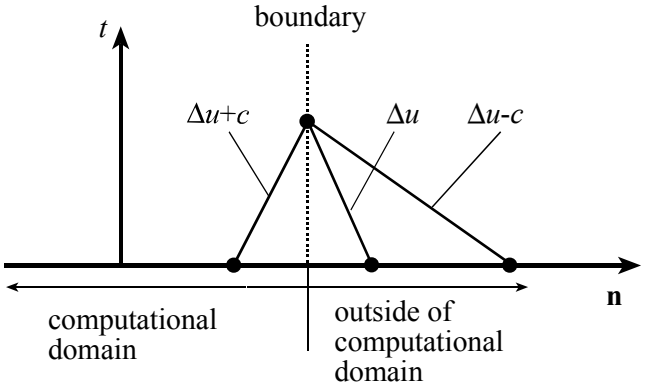


Figure 6.6 Illustration of the subsonic condition at inlet

When a flow is subsonic at an outlet, only one wave speed is negative. Therefore one value is expected on the boundary to represent the incoming wave (see the figure below). This condition is sometimes called an outlet control condition. Typically a pressure value is specified in most internal flows. Corresponding to the two positive waves, $\eta_1 = \eta_3 = 0$ are used.

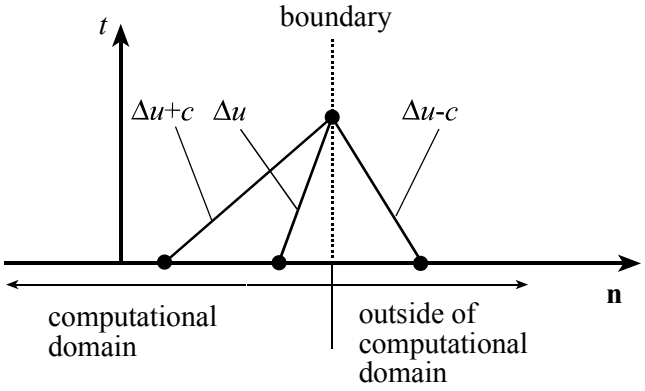


Figure 6.7 Illustration of the subsonic condition at outlet

No condition is required at the outlets where the flow becomes supersonic since all waves leave the computational domain. In this case, the boundary values are completely determined by the solution inside the domain (see the figure below).

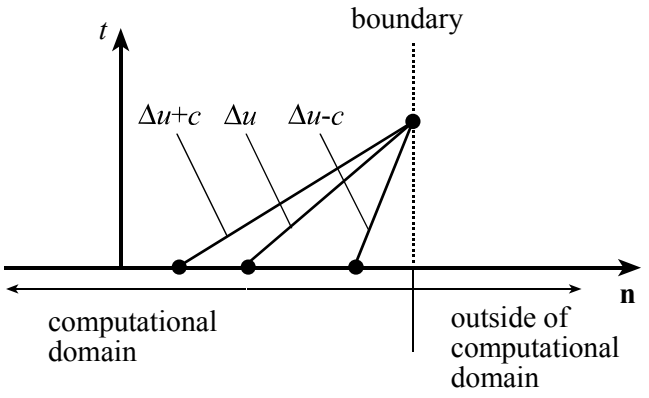


Figure 6.8 Illustration of the supersonic condition at outlet

In some situations where the required number of conditions cannot be fulfilled, a zero first derivative condition can be used to replace it. This is typically used along the outlet of subsonic external flows.

A typical set of fluid domain and associated boundary conditions for external flows is shown in the following figure, where the whole boundary S of the computational domain has been divided into the entrance $\overline{S}_e$ (where an external condition is applied), the boundary that bounds the solid body $\overline{S}_w$ (where a wall condition is applied) and the tail of the flow $\overline{S}_t$ (where a uniform condition is applied). The S_i here represents the interior part of that part of the boundary, while the over bar indicates that it contains the boundary of S_i as well. Therefore, the common part of S_i and S_j is the empty geometric set if $i \neq j$, while $\overline{S_i}$ and $\overline{S_j}$ share their common frontier where both conditions are applied. This boundary partition can be expressed as

$$\overline{S}_e \bigcup \overline{S}_w \bigcup \overline{S}_t = \overline{S} = S$$
$$S_i \bigcap S_j = \varnothing \qquad i \neq j; \quad i, j = e, w, t$$

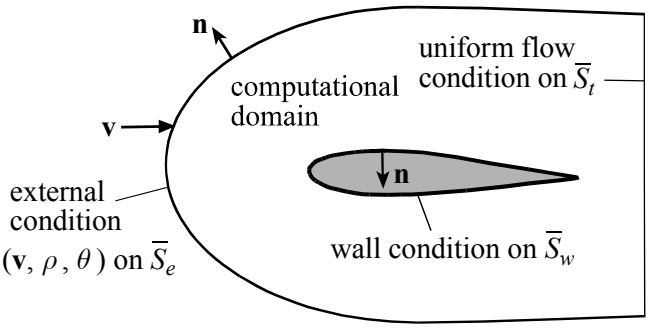


Figure 6.9 A typical boundary condition set for external high-speed compressible flows

A similar boundary condition set for internal flows is shown in the following figure, where the whole boundary S of the computational domain has been divided into the inlet boundary $\overline{S}_i$ (where an inlet condition is applied), the wall boundary $\overline{S}_w$ (where a wall condition is applied) and the outlet boundary $\overline{S}_o$ (where an outlet condition is applied). They also form the whole boundary and have no shared common part except the locations where two boundaries meet each other.

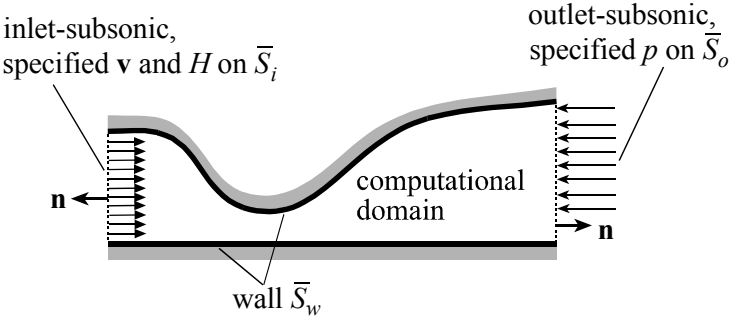


Figure 6.10 A typical fluid boundary condition set for internal high-speed compressible flows

From these conditions, we observe that:

- All parts of the boundary have been assigned one and only one fluid boundary condition (except on the interface of the sub-boundaries, where two conditions meet).
- The type of the condition follows the flow regime on the boundary.

This observation is a fundamental guideline for specifying boundary conditions in most high-speed compressible problems. Unlike the conditions for incompressible flows, there is no default condition that is implicitly applied to boundaries. Every boundary must be assigned a condition.

Some variables on boundaries can directly be imposed if they do not or only weakly depend on other solution variables. This type of condition is referred to as a usual boundary condition. It is applied to the governing equations in a simple manner. The conditions are either prescribed solution variables (where the variables are forced to be the prescribed values and the corresponding equations are removed) or forces, which are directly added to the fluid equations as external boundary forces. However, caution is necessary in using only the usual boundary conditions, since the set of properly defined boundary conditions should describe somewhere a complete flow condition rather than only a single variable. Therefore special conditions are strongly recommended in high-speed compressible flows.

Special boundary conditions may be more complicated or strongly depend on solution variables and additional "boundary" equations must be solved. They are discretized and assembled to the global system of equations. Special boundary conditions play important roles in high-speed compressible flows. In fact, the usual boundary conditions are rarely used.

Unless noted otherwise, special boundary conditions can only be applied to “geometries” of one dimension lower than the computational domain. More specifically, in 2D models, the “geometries” are lines or nodes that can be connected to lines and in 3D models, the “geometries” are surfaces or nodes that can be connected to surfaces. These “geometries” may be located on the boundary or inside the domain, depending on the type of boundary conditions.

Most boundary conditions impose only constraints on fluid variables (velocity, pressure, temperature, etc.). Some other conditions impose additional constraints on the boundary positions or boundary

displacements. These conditions are called kinematic conditions or moving boundary conditions. When a moving boundary condition is applied, the boundary nodal displacements are computed or specified. The interior nodal positions are adjusted in an "arbitrary" way to ensure the validity and quality of the mesh. In this case, an arbitrary Lagrangian-Eulerian (ALE) coordinate system is used in the fluid model.

6.4.2 Usual boundary conditions for fluid

6.4.2.1 Prescribed specific discharge

In this condition, a time-dependent specific discharge is directly prescribed

$$\rho v_i = \overline{\rho v}_i(t)$$

and applied to boundaries. The x_i-momentum equations at the boundary nodes are then replaced by this condition.

Be cautious when using this condition since there is no temperature condition applied yet. This boundary condition alone does not give a well-posed problem.

6.4.2.2 Zero specific discharge

Physically, this condition is equivalent to a prescribed specific discharge. However, this condition removes the degree of freedom of the specific discharge component.

Be cautious when using this condition since there is no temperature condition applied yet. This boundary condition alone does not give a well-posed problem.

6.4.2.3 Prescribed density

In this condition, a time-dependent density is directly prescribed

$$\rho = \overline{\rho}(t)$$

and applied to the boundaries. The continuity equations at boundary nodes are then replaced by this condition.

Be cautious when using this condition since there are no velocity and temperature conditions applied yet. This condition alone does not give a well-posed problem.

6.4.2.4 Concentrated force load

In this condition, a prescribed time-dependent force $\overline{\mathbf{F}}(t)$ is directly applied to the boundary. The force is added to the momentum equations of the boundary nodes. Note that the force can be directly applied to nodes. This force is also called an external nodal force.

This condition has no effect on the nodes where the specific discharge condition is prescribed. Recall that in this case the momentum equation has been replaced by the velocity condition.

Be cautious when using this condition. It functions like an additional force load to the momentum equation. It does not provide any information on the boundary variables. Therefore, to reach a well-posed boundary condition set, this condition should not be included in that set. In other words, another properly defined condition set must be applied to the same boundary.

6.4.2.5 Distributed normal-traction loads

This condition can only be applied to boundary lines and surfaces, respectively, of two-dimensional and three-dimensional computational domains. In this condition, a time-dependent normal stress (called normal-traction) $\overline{\tau}_{nn}(t) = \mathbf{n} \bullet \boldsymbol{\tau} \bullet \mathbf{n}$ is prescribed. The applied nodal force is then computed by

$$\mathbf{F}(t) = \int h^v \overline{\tau}_{nn}(t)\, d\mathbf{S}$$

where h^v is the virtual quantity of velocity on the boundary.

Note that the normal stress consists of the pressure and the normal shear stress. Along open boundaries, the normal shear stress is usually negligible compared with the pressure. Therefore, a normal-traction is usually applied to open boundaries where the pressure is known. In particular, when

$\bar{\tau}_{nn}(t) = 0$, the application of the normal-traction load is trivial because it can be reached by simply not specifying the condition.

This condition has no effect on the nodes where a normal specific discharge condition is prescribed. Recall that in this case, the normal momentum equation has been replaced by the normal specific discharge condition.

Similar to a concentrated force condition, caution is necessary when using this condition. It functions like an additional force load to the normal momentum equation. It does not provide any information on the boundary variables. Therefore, to reach a well-posed boundary condition set, this condition should not be included in that set. In other words, another properly defined condition set must be applied to the same boundary.

6.4.2.6 Field centrifugal loads

As described in Section (2.3), when the whole field is rotating, the flow equations can be formulated in a rotational reference coordinate system. The problem is then equivalent to a problem defined in a fixed coordinate system subjected to additional centrifugal forces. These forces can be written as

$$\mathbf{F}(t) = \int h^v \mathbf{f}_c \, dV$$

where h^v is the virtual quantity of velocity and $\mathbf{f}_c$ is defined in Chapter 2 and rewritten here, for the purpose of indicating the prescribed parameters in the force

$$\begin{aligned}\mathbf{f}_c = &-\rho\ddot{\bar{\mathbf{x}}}_0(t) - \rho\dot{\bar{\boldsymbol{\Omega}}}(t)\times(\mathbf{x}-\bar{\mathbf{x}}_0(t)) - 2\bar{\boldsymbol{\Omega}}(t)\times(\rho\mathbf{v}) \\ &- \rho\bar{\boldsymbol{\Omega}}(t)\times\bar{\boldsymbol{\Omega}}(t)\times(\mathbf{x}-\bar{\mathbf{x}}_0(t))\end{aligned}$$

It must be kept in mind that the velocity solved in this coordinate system is the relative velocity with respect to the rotational frame (which is denoted as $\mathbf{v}_r$ in Section 2.3).

Strictly speaking, this condition is not a "boundary" condition since the force is applied to the whole domain.

6.4.2.7 Prescribed total energy

In this condition, a time-dependent total energy can be directly prescribed and applied to boundaries:

$$\rho E = \overline{\rho E}(t)$$

The energy equations at the boundary nodes are then replaced by this condition.

Be cautious when using this condition since there are no density and velocity conditions applied yet. If only this condition is applied to a boundary, the problem may not be well-posed.

6.4.2.8 Concentrated heat flow load

In this condition, a prescribed time-dependent heat flow load $\bar{Q}(\mathrm{t})$ is directly applied to the boundaries. The load is added to the energy equations of the boundary nodes. Note that the heat flow loads can directly be applied to nodes.

This condition has no effect if applied to the boundary where a total energy is prescribed. Recall that in this case, the energy equation has been replaced by the prescribed total energy condition.

Be cautious when using this condition. It functions like an additional heat flow load to the energy equation. It does not provide any information on the boundary variables. Therefore, to reach a well-posed boundary condition set, this condition should not be included in that set. In other words, another properly defined condition set must be applied to the same boundary.

6.4.2.9 Distributed heat flux load

The condition can only be applied to boundary lines and surfaces, of two-dimensional and three-dimensional computational domains, respectively. Distributed heat flux load is equivalent to a concentrated force load obtained by specifying the normal component of the external heat flux $\hat{q}_n(t)$ imposed onto the fluid domain. The applied heat flow is then computed by

$$Q(t) = \int h^{\theta} \overline{q}_n(t)\, dS$$

where h^{θ} is the virtual quantity of temperature on the boundary. In particular, when $\overline{q}_n(t) = 0$, the application of the heat flux is trivial because it can be reached by simply not specifying the condition.

Be cautious when using this condition. It functions only like an additional heat flux load to the energy equation. It does not provide any information on the boundary variables. Therefore, to reach a well-posed boundary condition set, this condition should not be included in that set. In other words, another properly defined condition set must be applied to the same boundary.

6.4.3 Special boundary conditions

A special boundary condition is a combination of a few boundary values specified together. Every value can either represent a Dirichlet condition (prescribed variable value), or a Neumann condition giving the zero value of first derivative. Let us denote any variable by ϕ ($\phi = u, p, \theta, H, e, \ldots$). We use the following notation to represent the possible conditions

$$\phi = \overline{A}(t) + \overline{B}(t)\phi_i$$

where ϕ_i is the value of the same field solution variable adjacent to the boundary.

When the variable is prescribed, choose $\overline{B}(t)$ =0 and specify $\overline{A}(t)$ to be the prescribed value of the variable.

On the other hand, when a zero Neumann condition is applied, choose $\overline{A}(t)$ =0 and $\overline{B}(t)$ =1. This means that a first order approximation to the Neumann condition will be used.

This notation is used for most of the variables that are required in the special boundary conditions for high-speed compressible flows.

6.4.3.1 Fixed wall

In the interface of a fluid and a fixed solid, no-slip or slip conditions are usually applied. On fixed wall conditions, the boundary is fixed. In other words, the boundary displacement is zero. Fixed wall conditions can only be applied to boundary lines and surfaces of two-dimensional and three-dimensional computational domains, respectively.

No-slip condition

When a no-slip condition on fixed walls is applied, the fluid velocity vector on that wall is prescribed to be zero

$$\mathbf{v} = \mathbf{0}$$

For the temperature, two types of conditions are usually considered: either the wall temperature is prescribed ($\theta = \bar{\theta}(t)$) or an adiabatic wall condition is applied ($\mathbf{n} \cdot \tilde{\mathrm{N}}\theta = 0$). These two conditions can be specified by choosing $(\bar{A}, \bar{B}) = (\bar{\theta}(t), 0)$ and $(\bar{A}, \bar{B}) = (0,1)$ respectively.

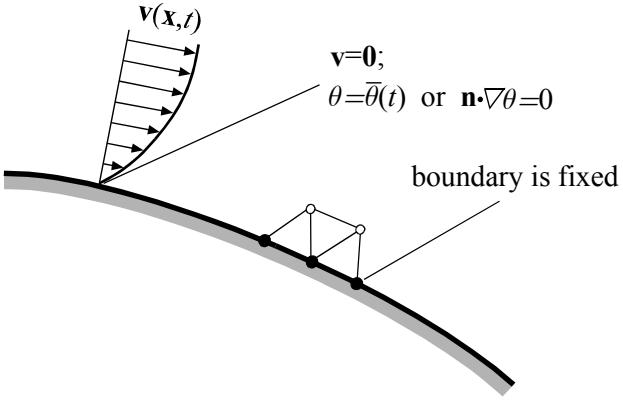


Figure 6.11 No-slip condition on fixed walls for high-speed compressible flows

This condition is usually applied to wall boundaries in viscous flows.

Slip condition

In this condition, the normal component of the velocity vector is prescribed to be zero

$$\mathbf{v} \cdot \mathbf{n} = 0$$

while the tangential components are free and computed as unknown variables through the governing equations.

The temperature condition can be either a prescribed temperature or a specified zero heat flux, in the same manner as in the no-slip condition.

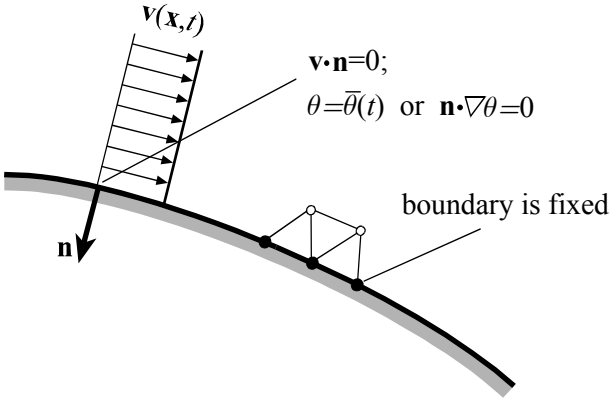


Figure 6.12 Slip condition on fixed walls for high-speed compressible flows

This condition is usually applied to wall boundaries when the Euler equations are solved (i.e., when the viscous effects are negligible).

6.4.3.2 Angular velocity

The condition is used to prescribe all velocity components in a special manner: by means of a time-dependent angular velocity $\bar{\mathbf{\Omega}}(t)$ (see figure below). This condition can be applied to fixed or moving boundaries, where the coordinates of the boundary nodes vary with time. The fluid velocity is then computed as

$$\mathbf{v}(t) = \bar{\boldsymbol{\Omega}}(t) \times \left(\mathbf{x}(t) - \bar{\mathbf{x}}_0 \right)$$

where $\bar{\mathbf{x}}_0$ is the center of rotation and $\mathbf{x}(t)$ denotes the boundary coordinates.

The temperature condition can be prescribed or the zero heat flux is specified, in the same manner as in the no-slip wall condition.

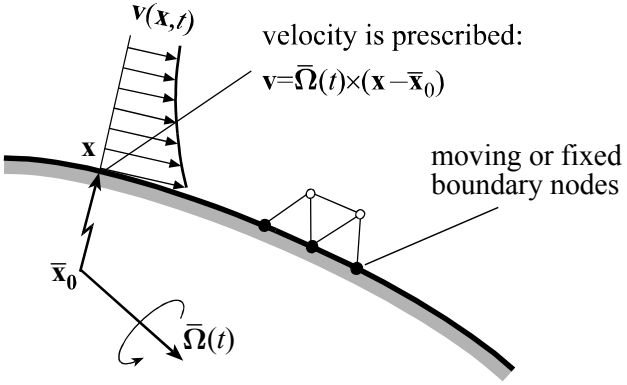


Figure 6.13 Angular velocity condition for high-speed compressible flows

6.4.3.3 Moving wall

This condition broadens applications of the fixed wall condition to moving wall conditions. In this condition, the boundary displacement $\bar{\mathbf{d}}(t)$ is prescribed while the fluid velocity is computed based on the displacement. This implicitly applies the Lagrange formulation to fluid boundary nodes. A moving wall condition is therefore a moving boundary condition. This condition can only be applied to boundary lines and surfaces of two-dimensional and three-dimensional computational domains, respectively.

No-slip condition on moving walls

The no-slip condition on a wall implies a consistency between the fluid velocity and the solid velocity

$$\mathbf{v} - \dot{\bar{\mathbf{d}}}(t) = \mathbf{0}$$

The temperature condition can be either a prescribed temperature or a specified zero heat flux, in the same manner as in the no-slip condition.

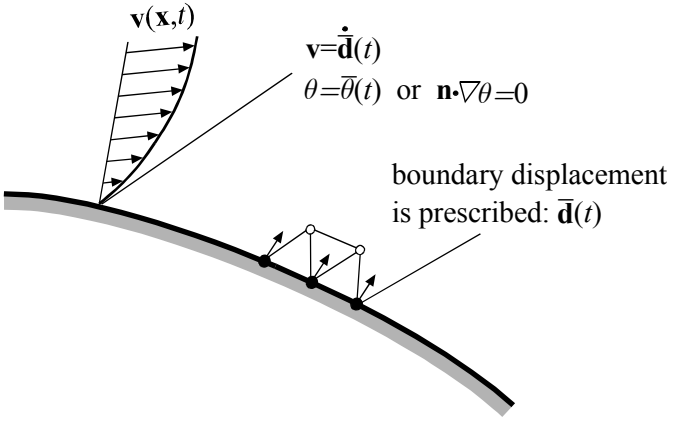


Figure 6.14 No-slip condition on moving walls for high-speed compressible flows

This condition is usually applied to moving wall boundaries in viscous flows. The dominant motion of the moving wall must be in the normal direction of the boundary. If the tangential movement is large, the moving wall conditions with specified tangential or rotational velocity should be more adequate.

Slip condition on moving walls

In slip wall conditions, only the normal component of the velocity is enforced to be consistent with the solid counterpart

$$\left(\mathbf{v}-\dot{\bar{\mathbf{d}}}(t)\right)\bullet\mathbf{n}=0$$

while the tangential components are free and computed as unknown variables from the governing equations.

The temperature condition can be either a prescribed temperature or a specified zero heat flux, in the same manner as in the no-slip condition.

This condition is usually applied to moving wall boundaries where the viscous effects are negligible.

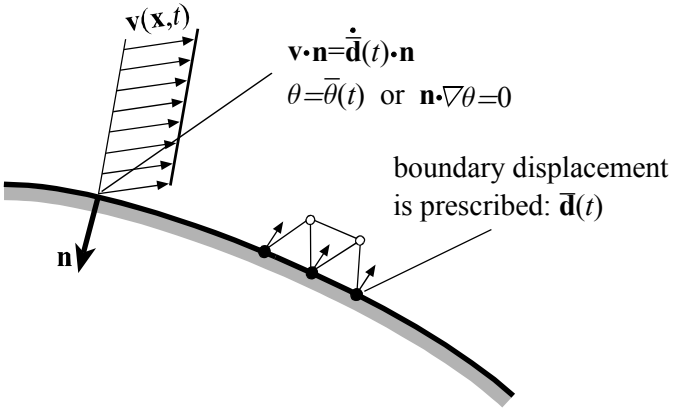


Figure 6.15 Slip condition on moving walls for high-speed compressible flows

In no-slip and slip moving wall conditions, the Lagrangian coordinate system has been implicitly applied to both the normal and tangential directions of moving boundaries. These conditions can present difficulties with moving meshes if the tangential displacement of the moving wall is large. The use of a Lagrangian coordinate system in the tangential direction results in largely skewed meshes. On the other hand, the use of the Lagrangian formulation to that direction is not absolutely necessary. Most tangential movements do not change the shape of computational fluid domains. Therefore, an ALE coordinate system can be used. In particular, the displacement in the tangential direction can be completely avoided if the Eulerian formulation is applied to that direction.

In this ALE formulation, both the tangential velocity and the displacement vector are prescribed. They may or may not be consistent in the tangential direction. When they are consistent, the Lagrangian formulation is implicitly used. When the tangential displacement is zero, the Eulerian formulation is then specified.

The new formulation is not only advantageous for some cases to reach less distorted meshes, but is also necessary for other cases where otherwise a valid mesh cannot be maintained. Two such formulations are introduced here.

No-slip condition of moving walls (type=tangential)

In this condition, the displacement $\bar{\mathbf{d}}(t)$ is prescribed. This displacement causes the boundary to move. The normal fluid velocity is then computed based on the normal displacement. The tangential velocity is prescribed separately (see the next figure). This condition is

$$\mathbf{v} = \bar{v}(t)\boldsymbol{\tau} + \dot{\bar{\mathbf{d}}}(t)\bullet(\mathbf{I} - \boldsymbol{\tau}\boldsymbol{\tau})$$

where $\bar{v}(t)$ is the prescribed tangential velocity and $\boldsymbol{\tau}$ is the tangential direction computed by means of a specified direction $\bar{\mathbf{a}}$ and the normal direction on the boundary $\mathbf{n}$

$$\boldsymbol{\tau} = \bar{\mathbf{a}} \times \mathbf{n}$$

Notice here that the tangential direction is not directly required, since it is, in general, not a constant vector.

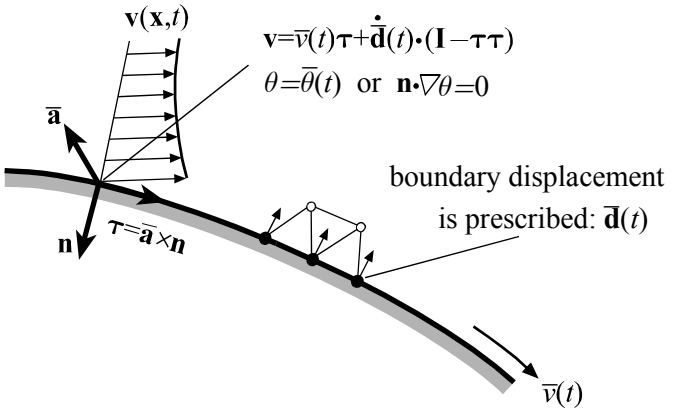


Figure 6.16 Moving wall condition (type=tangential) for high-speed compressible flows

The temperature condition can be either a prescribed temperature or a specified zero heat flux, in the same manner as in the no-slip condition.

No-slip condition of moving walls (type=rotational)

When the dominant boundary motion is determined by a rotation (see the next figure), we can apply an ALE formulation to that direction while we use the Lagrangian coordinate system for the other directions. In this case, the displacement vector and the angular velocity is prescribed. The velocity vector is then computed as

$$\mathbf{v}(t) = \bar{\mathbf{\Omega}}(t) \times (\mathbf{x}(t) - \bar{\mathbf{x}}_0) + \dot{\bar{\mathbf{d}}}(t) \bullet (\mathbf{I} - \boldsymbol{\tau}\boldsymbol{\tau})$$

where $\bar{\mathbf{x}}_0$ is the center coordinate of the rotation, $\bar{\mathbf{\Omega}}$ is the angular velocity of the solid wall and the tangential direction $\boldsymbol{\tau}$ is computed as

$$\boldsymbol{\tau} = \frac{\bar{\mathbf{\Omega}}(t) \times (\mathbf{x}(t) - \bar{\mathbf{x}}_0)}{\left\| \bar{\mathbf{\Omega}}(t) \times (\mathbf{x}(t) - \bar{\mathbf{x}}_0) \right\|}$$

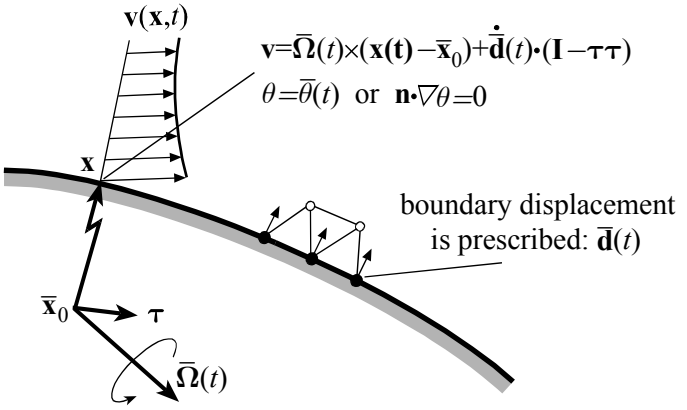


Figure 6.17 Moving wall condition (type=rotational) for high-speed compressible flows

The temperature condition can be either a prescribed temperature or a specified zero heat flux, in the same manner as in the no-slip condition.

6.4.3.4 External flow

Three boundary values and the direction of the specified velocity are required in this condition. These values can be $(\bar{\rho}(t), \bar{v}(t), \bar{\mathbf{M}}(t))$ or $(\bar{\theta}(t), \bar{\rho}(t), \bar{v}(t))$, where $\bar{v}$ is the magnitude of the velocity in a specified flow direction $\bar{\mathbf{m}}$.

In addition, a velocity condition imposed on the component in the direction $\mathbf{k}$ can also be specified, where $\mathbf{k}$ is tangential to $\bar{\mathbf{m}}$. Let $v_k \equiv \mathbf{v} \bullet \mathbf{k}$ and the coordinate in the normal direction $\mathbf{n}$ be n, this condition can then be either $v_k = 0$ (represented as $\bar{s}$ =0) or $\partial v_k / \partial n = 0$ (represented as $\bar{s}$ =1).

Based on these conditions, we can obtain a complete description of the conservative variables.

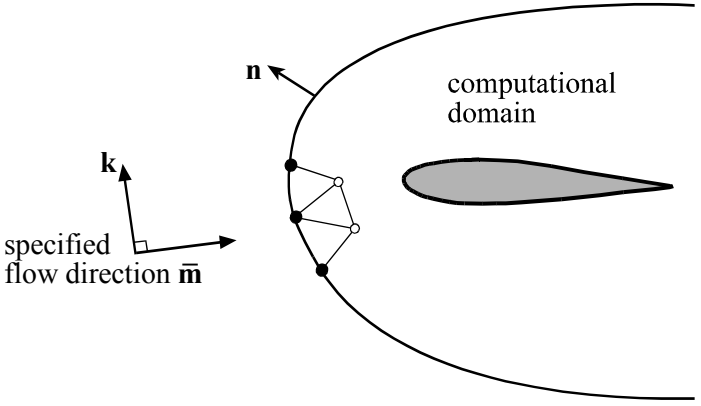


Figure 6.18 External flow condition

In the case that ($\overline{\rho}(t), \overline{v}(t), \overline{\mathrm{M}}(t)$) is specified, we have

$$\mathbf{U} = \begin{bmatrix} \overline{\rho} \\ \overline{\rho}(\overline{v}\,\overline{\mathbf{m}} + \overline{s}v_i\mathbf{k}) \\ \frac{1}{2}\overline{\rho}(\overline{v}^2 + \overline{s}v_i^2) + \overline{\rho}\,\overline{v}^2/\overline{\mathrm{M}}^2\alpha\gamma \end{bmatrix}$$

In the case that ($\overline{\theta}(t), \overline{\rho}(t), \overline{v}(t)$) is specified, we have

$$\mathbf{U} = \begin{bmatrix} \overline{\rho} \\ \overline{\rho}(\overline{v}\,\overline{\mathbf{m}} + \overline{s}v_i\mathbf{k}) \\ \frac{1}{2}\overline{\rho}(\overline{v}^2 + \overline{s}v_i^2) + C_v\overline{\rho}\overline{\theta} \end{bmatrix}$$

This condition can only be applied to boundary lines and surfaces of two-dimensional and three-dimensional computational domains, respectively.

6.4.3.5 Supersonic at inlet

Three boundary values are required in this condition. These values can be ($\bar{\rho}(t),\bar{u}(t),\bar{\mathrm{M}}(t)$), ($\bar{p}(t),\bar{\theta}(t),\bar{u}(t)$) or ($\bar{p}(t),\bar{u}(t),\bar{\mathrm{M}}(t)$), where $\bar{u}$ is the magnitude of the velocity in the normal direction of the boundary $\mathbf{n}$.

In addition, a velocity condition imposed on the component in the tangential direction $\boldsymbol{\tau}$ can also be specified. Let $v_\tau \equiv \mathbf{v}\bullet\boldsymbol{\tau}$ and the coordinate in the direction $\mathbf{n}$ be n, this condition can be either $v_\tau = 0$ (represented as $\bar{s}$ =0) or $\partial v_\tau / \partial n = 0$ (represented as $\bar{s}$ =1).

Based on these conditions, we can obtain a complete description of the conservative variables.

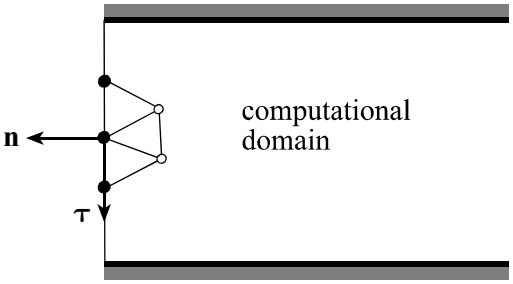


Figure 6.19 Sketch for inlet and outlet flow conditions

In the case that ($\bar{\rho}(t),\bar{u}(t),\bar{\mathrm{M}}(t)$) is specified, we have

$$\mathbf{U} = \begin{bmatrix} \bar{\rho} \\ \bar{\rho}(\bar{u}\mathbf{n} + \bar{s}v_i\boldsymbol{\tau}) \\ \frac{1}{2}\bar{\rho}(\bar{u}^2 + \bar{s}v_i^2) + \bar{\rho}\,\bar{u}^2/\bar{\mathrm{M}}^2\alpha\gamma \end{bmatrix}$$

In the case that ($\bar{p}(t),\bar{\theta}(t),\bar{u}(t)$) is specified, we have

$$\mathbf{U} = \begin{bmatrix} \overline{p}/(C_p - C_v)\overline{\theta}\,(\equiv \rho) \\ \rho(\overline{u}\mathbf{n} + \overline{s}v_i\boldsymbol{\tau}) \\ \frac{1}{2}\rho(\overline{u}^2 + \overline{s}v_i^2) + C_v\rho\overline{\theta} \end{bmatrix}$$

In the case that ($\overline{p}(t), \overline{u}(t), \overline{\mathrm{M}}(t)$) is specified, we have

$$\mathbf{U} = \begin{bmatrix} \gamma\overline{p}\overline{\mathrm{M}}^2/\overline{u}^2\,(\equiv \rho) \\ \rho(\overline{u}\mathbf{n} + sv_i\boldsymbol{\tau}) \\ \frac{1}{2}\rho(\overline{u}^2 + sv_i^2) + \overline{p}/\alpha \end{bmatrix}$$

This condition can only be applied to boundary lines and surfaces of two-dimensional and three-dimensional computational domains, respectively.

6.4.3.6 Subsonic at inlet

Two boundary values are required in this condition. They control the flow condition in the normal direction to the boundary. Thus the velocity specified in this condition is the magnitude of the velocity in the normal direction $\mathbf{n}$. Since the inlet flow condition is assumed to be subsonic, only one wave speed (d_1) is positive. Therefore $\eta_1 = 0$ is assumed.

In addition, a velocity condition imposed on the component in the tangential direction $\boldsymbol{\tau}$ can also be specified. Let $v_\tau \equiv \mathbf{v}\bullet\boldsymbol{\tau}$ and the coordinate in the direction $\mathbf{n}$ be n, this condition can be either $v_\tau = 0$ (represented as $\overline{s}$ =0) or $\partial v_\tau/\partial n = 0$ (represented as $\overline{s}$ =1).

For the purpose of convenience in the input, a few combinations of these values can be used. They are

$$(p,H),(p,\theta),(\theta,u),(\theta,\rho u),(\rho,u),(H,u)$$

This condition can only be applied to boundary lines and surfaces of two-dimensional and three-dimensional computational domains, respectively.

6.4.3.7 Subsonic at outlet

One boundary value is required in this condition. It controls the flow condition in the normal direction of the boundary. Thus the velocity specified in this condition is the magnitude of the velocity in the normal direction $\mathbf{n}$. Since the outlet flow condition is assumed to be subsonic, two wave speeds (d_1 and d_3) are positive and therefore $\eta_1 = \eta_3 = 0$ are assumed.

In addition, a velocity condition imposed on the component in the tangential direction $\boldsymbol{\tau}$ can be specified. Let $v_\tau \equiv \mathbf{v} \bullet \boldsymbol{\tau}$ and the coordinate in the direction $\mathbf{n}$ be n, this condition can be either $v_\tau = 0$ (represented as $\overline{s}$ =0) or $\partial v_\tau / \partial n = 0$ (represented as $\overline{s}$ =1).

Most frequently a pressure condition is specified, although other conditions are also applicable. The available boundary values that can be specified are p, θ, ρ, and e.

This condition can only be applied to boundary lines and surfaces of two-dimensional and three-dimensional computational domains, respectively.

6.4.3.8 Supersonic at outlet

When an outflow becomes supersonic, all wave speeds are positive, so all η_i (i=1,2,3,4,5) are zero. This indicates that no boundary values are required under this flow condition. The solutions of $\eta_i = 0$ are simply equivalent to $\Delta\mathbf{U} = \mathbf{0}$. This is the same as specifying zero first derivatives for all solution variables.

Although this type of condition is suitable for supersonic flows at outlet, it is also frequently used as a “uniform” flow condition where all variables are extrapolated.

This condition can only be applied to boundary lines and surfaces of two-dimensional and three-dimensional computational domains, respectively.

6.4.3.9 Symmetry boundary conditions

This condition is usually applied to the boundary where the flows on both sides are symmetric (another side is not modeled). The conditions are

$$u = 0, \quad \frac{\partial f}{\partial n} = 0 \quad \left(f = p, \theta \text{ and } \mathbf{v} - u\mathbf{n} \right)$$

No parameters are required in this condition.

This condition can only be applied to boundary lines and surfaces of two-dimensional and three-dimensional computational domains, respectively.

6.4.3.10 Fluid-structure interface

The velocity and pressure conditions on a fluid-structure interface are the same as for incompressible flows (see Section 3.4 for details). The temperature condition can be either prescribed temperature or the zero heat flux is specified in the same manner as in the no-slip wall condition.

This condition can only be applied to boundary lines and surfaces of two-dimensional and three-dimensional computational domains, respectively.

6.4.3.11 Gap

The velocity and pressure conditions are the same as in an incompressible flow case (see Section 3.4 for details). If the status of the gap is closed, an adiabatic wall is further assumed.

This condition can only be applied to boundary lines and surfaces of two-dimensional and three-dimensional computational domains, respectively.

6.4.4 User-supplied boundary condition

The application of this condition is the same as in an incompressible flow case (see Section 3.4) for details. The only difference is that conservative variables are used here. That is, in the case $I_{\text{var}} = 1$, $\mathbf{\Phi}$ is defined as $(\rho u, \rho v, \rho w, \rho, \rho E)$.

6.5 Initial conditions

Although the conservative variables are used in the solutions, the primitive variables are still used in specifying initial conditions for the convenience of input.

Recalling that pressure, temperature and density must satisfy the state equation represented by the ideal gas law, zero pressure or zero temperature must not be specified. Since the default initial conditions to all variables are zero, actual initial conditions are always required in both transient and steady-state analyses. Good initial conditions, as will be seen in Chapter 13, may accelerate the convergence during equilibrium iterations. In certain cases, good initial conditions may become the key factor in obtaining converged solutions.

Initial conditions can be improved in restart analyses or using the solutions mapped from similar solutions.

6.6 Material models

6.6.1 Constant material model

This is the simplest yet most frequently used material model. In this model, all fluid properties are assumed to be constant. They are

μ = fluid viscosity
μ_2^* = ratio of the second viscosity to the viscosity
C_p = specific heat at constant pressure
C_v = specific heat at constant volume
k = thermal conductivity
q^B = rate of heat generated per unit volume

6.6.2 Power-law model

This model contains the modifications of the viscosity and the heat conductivity defined in Sutherland's formulae

$$\mu = \mu_0 \left(\frac{\theta}{\theta_\mu} \right)^{m_\mu} \frac{\theta_\mu + S_\mu}{\theta + S_\mu}, \quad k = k_0 \left(\frac{\theta}{\theta_k} \right)^{m_k} \frac{\theta_k + S_k}{\theta + S_k}$$

The required fluid properties are

μ_0 = fluid viscosity at temperature θ_μ
θ_μ = reference temperature used for calculating μ
S_μ = constant temperature used for calculating μ
m_μ = exponent used for calculating μ
k_0 = thermal conductivity at temperature θ_k
θ_k = reference temperature used for calculating k
S_k = constant temperature used for calculating k
m_k = exponent used for calculating k
μ_2^* = ratio of the second viscosity to the viscosity
C_p = specific heat at constant pressure
C_v = specific heat at constant volume
q^B = rate of heat generated per unit volume

6.6.3 Temperature-dependent model

This model contains the same parameters as those in the power-law model. However, the following parameters in this model depend on temperature

$$\mu_0,\ C_p,\ C_v,\ k_0,\ q^B = f(\theta)$$

6.6.4 Pressure-dependent model

This model contains the same parameters as those in the power-law model. However, the following parameters in this model depend on pressure

$$\mu_0,\ C_p,\ C_v,\ k_0,\ q^B = f(p)$$

6.6.5 Pressure-temperature-dependent model

This model contains the same parameters as those in the power-law model. However, the following parameters in this model depend on pressure and temperature

$$\mu_0,\ C_p,\ C_v,\ k_0,\ q^B = f(p,\theta)$$

The input of these data assumes the following format:

$$\begin{Bmatrix} f(p_1,\theta_{11}) & f(p_1,\theta_{12}) & \cdots & f(p_1,\theta_{1n_1}) \\ f(p_2,\theta_{21}) & f(p_2,\theta_{22}) & \cdots & f(p_2,\theta_{2n_2}) \\ \cdots & \cdots & \cdots & \cdots \\ f(p_m,\theta_{m1}) & f(p_m,\theta_{m2}) & \cdots & f(p_m,\theta_{mn_k}) \end{Bmatrix}$$

where m is the number of pressure points and n_i $(i = 1,2,...,m)$ is the number of temperature points at the pressure point p_i.

6.6.6 User-supplied materials

In this model, the user can specify the material data through a user-supplied subroutine, which is designed specifically for high-speed compressible flows. The subroutine will be invoked whenever material data are required in the computations. Current pressure and temperature will be passed in as arguments, while the program returns the following fluid properties

$$\mu, k, q^B, f, \frac{\partial f}{\partial p}, \frac{\partial f}{\partial \theta} \quad \left(f = C_v \,\&\, C_p\right)$$

The following example demonstrates the use of a user-supplied material data set for $\mathrm{Re} = 1000$, $\gamma = 1.4$, $\mathrm{Pr} = 0.71$ and the definitions

$$\mu = \frac{1}{\mathrm{Re}}, \quad C_v = \frac{1}{\gamma - 1}, \quad C_p = 1 + C_v, \quad k = \frac{\mu C_p}{\mathrm{Pr}}, \quad q^B = 0$$

```
      SUBROUTINE CFVUSR(P,T,CMAT)
C=======================================================================
C
C     USER-SUPPLIED MATERIAL DATA FOR COMPRESSIBLE FLUID FLOWS
C
C     INPUT : P,T
C     OUTPUT: CMAT
C
C     ----------------------------------------------------------------
C     WHERE
C        P       = PRESSURE AT THE LOCATION
C        T       = TEMPERATURE AT THE LOCATION
C        CMAT(1) = SPECIFIC HEAT AT CONSTANT PRESSURE           (CP )
C        CMAT(2) = DERIVATIVE OF CMAT(1) WITH RESPECT TO P      (CPP)
C        CMAT(3) = DERIVATIVE OF CMAT(1) WITH RESPECT TO T      (CPT)
C        CMAT(4) = SPECIFIC HEAT AT CONSTANT VOLUME             (CV )
C        CMAT(5) = DERIVATIVE OF CMAT(4) WITH RESPECT TO P      (CVP)
C        CMAT(6) = DERIVATIVE OF CMAT(4) WITH RESPECT TO T      (CVT)
C        CMAT(7) = VISCOSITY
C        CMAT(8) = COEFFICIENT OF THERMAL CONDUCTIVITY
C        CMAT(9) = RATE OF HEAT GENERATED PER UNIT VOLUME
C
C        BE NOTED THAT THE STATE EQUATIONS ARE DEFINED AS
C
C              P = (CP-CV)*D*T
C              T =  E/CV
C
C        WHERE D =  DENSITY
C              E =  INTERNAL ENERGY
C
C=======================================================================
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION  CMAT(*)
C-----------------------------------------------------------------------
      P  =  P
      T  =  T
C-----------------------------------------------------------------------
C
C     FOR A PERFECT GAS, USING T_SCALE = (U_SCALE)**2/(CP-CV), THE
C     MATERIAL DATA ARE REPRESENTED BY REYNOLDS NUMBER "RE", PRANDTL
C     NUMBER "PR" AND THE RATIO OF SPECIFIC HEATS "GAMMA".
C
C       XMU   =  1/RE
```

```
C       XKCON =  XMU*CP/PR
C       CV    =  1/(GAMMA-1)
C       CP    =  1 + CV
C
C-----------------------------------------------------------------------
      RE      =  1000.D0
      PR      =  0.71D0
      GAMMA   =  1.4D0
C
      XMU     =  1.D0/RE
      CV      =  1.D0/(GAMMA-1.D0)
      CP      =  1.D0 + CV
      CMAT(1) =  CP
      CMAT(2) =  0.D0
      CMAT(3) =  0.D0
      CMAT(4) =  CV
      CMAT(5) =  0.D0
      CMAT(6) =  0.D0
      CMAT(7) =  XMU
      CMAT(8) =  XMU*CP/PR
      CMAT(9) =  0.D0
C-----------------------------------------------------------------------
C     FOR A REAL GAS, SUPPLY THE FUNCTIONS HERE AND COMMENT OUT THE
C     UPPER SECTION
C-----------------------------------------------------------------------
 9999 CONTINUE
      RETURN
      END
```

Chapter 7 Formulation of turbulence in high-speed compressible flows

In this chapter, we introduce our two-equation turbulence model for high-speed compressible flows.

The model introduced here can be coupled with ADINA solid models in fluid-structure interaction analyses. They can also be associated with mass transfer analyses.

The following table presents a quick overview of the capabilities that are available for this turbulence model.

Table 7-1 Functionalities in the K-ε turbulence model for high-speed compressible flows

Category	**Functionality**
Coupled models Note: Heat transfer must be coupled.	ADINA solid model
	solid element groups
	heat transfer
	mass transfer
Computational domains	2D planar, 2D axisymmetric and 3D
Analyses	steady-state and transient
Elements	3-node triangle
	4-node tetrahedron
Material model	turbulence K-ε
Initial conditions Note: Zero pressure or temperature cannot be specified.	zero (default conditions)
	specified
	mapped from other solutions
	restart run

Boundary conditions Note: The column furthest to the right represents the solution variables or equations that are immediately affected by the specified condition. The solution variable **d** indicates a moving boundary condition.	prescribed specific discharges	$\rho\mathbf{v}$
	zero specific discharges	
	prescribed density	
	concentrated force load	
	distributed normal-traction load	
	field centrifugal load	
	prescribed total energy	ρE
	concentrated heat flow load	
	distributed heat flux load	
	fixed wall	$\mathbf{v}, \theta, K, \varepsilon$
	angular velocity	$\mathbf{v}, \theta$
	moving wall	$\mathbf{v}, \theta, K, \varepsilon, \mathbf{d}$
	external flow condition	$\rho, \rho\mathbf{v}, \rho E$
	supersonic at inlet	
	subsonic at inlet	
	supersonic at outlet	
	subsonic at outlet	
	symmetric	
	fluid-structure interface	$\mathbf{v}, \theta, K, \varepsilon, \mathbf{d}$
	gap	all
	user-supplied	
	prescribed turbulence K-ε	K, ε
	zero flux of turbulence variables	
Solvers for linearized equations	Gauss elimination method (COLSOL)	
	sparse solver	
	iterative methods (RPBCG, RPGMRES, and multi-grid)	
Other capabilities	automatically nondimensional procedure	
	automatic time-stepping CFL option	
	automatic time-stepping ATS option	
	skew system	
	constraint condition	
	conjugate heat transfer	
	pressure datum	

7.1 Governing equations

The governing equations for the fluid flow are those of high-speed compressible flows. However, the viscosity and heat conductivity are modified to include the turbulence effect and the K-ε model is used to model the required variables. These equations are listed in Table 2-1. Note that the fluid solution variables are modified to include the effect of compressibility. The solution variables are $(\rho, \rho\mathbf{v}, \rho E, K, \varepsilon)$. In the case

of moving mesh problems, the ALE formulation described in Section 2.14 is used.

7.2 Numerical method

The time integration is performed as for the fluid flow equations. The fluid equations and their discretization are described in Section 6.2.

The finite element method is also used to discretize the governing equations of the turbulence solution variables. The finite element equations are obtained by establishing a weak form of the turbulence equations using the Galerkin procedure. The equations of K and ε are weighted with the virtual quantities of K and ε, respectively. The equations are integrated over the computational domain V. The divergence theorem is used to lower the order of the derivatives of turbulence fluxes.

The variational forms of the K-ε equations are

$$\int_V \left(h^f G^f + \mathbf{Q}^f \bullet \nabla h^f \right) dV = 0$$

where f represents K and ε, h^K and h^ε are virtual quantities of K and ε respectively, and

$$G^K = y^a \left(\rho \frac{\partial K}{\partial t} + \rho \mathbf{v} \bullet \nabla K - S_K \right)$$

$$G^\varepsilon = y^a \left(\rho \frac{\partial \varepsilon}{\partial t} + \rho \mathbf{v} \bullet \nabla \varepsilon - S_\varepsilon \right)$$

$$\mathbf{Q}^K = y^a \mathbf{q}_K$$

$$\mathbf{Q}^K = y^a \mathbf{q}_K$$

7.3 Elements

The elements that can be used for high-speed compressible turbulence variables are the same as those for laminar model of high-speed compressible flows.

7.4 Boundary conditions

The boundary conditions for fluid variables are usually the same as for laminar models, although the treatment of the wall condition or its analogue is modified as explained later in this section. The conditions introduced here are only for the turbulence variables K and ε.

A typical turbulence condition may be prescribed as a turbulence variable at the inlet, zero first derivative of the turbulence variables at the outlet and a solid wall condition. In general, all parts of the boundary have to be applied with one and only one turbulence boundary condition. There is at least one wall condition or a prescribed turbulence variable condition in steady-state analyses. The available conditions are introduced here.

7.4.1 Prescribed turbulence K-ε

prescribed turbulence K-ε

In this condition, time-dependent turbulence variables are directly prescribed

$$K = \overline{K}(t), \quad \varepsilon = \overline{\varepsilon}(t)$$

and applied to boundaries. The K and ε equations at the boundary nodes are then replaced by these equations.

In addition to directly given values, they can also be prescribed indirectly via

$$K = \tfrac{3}{2}\left(\bar{i}\,\bar{v}\right)^2$$
$$\varepsilon = K^{\frac{3}{2}} \Big/ \left(0.3\bar{L}\right)$$

where, $\bar{v}, \bar{L}$ and $\bar{i}$ are velocity scale, length scale and turbulence intensity respectively.

This condition is usually applied to inlet boundaries.

7.4.2 Zero flux of turbulence variables

The natural boundary condition of any equation requires a specified normal flux of that variable. The turbulence flux condition is currently unavailable in ADINA-F (practically such conditions are not available). Therefore, a zero flux is implicitly assumed along the boundary where no turbulence condition is explicitly specified.

7.4.3 Solid walls

This condition is probably the most important one in modeling turbulence. At a solid wall the no-slip condition applies so that both the mean and fluctuating velocities are zero. In the near wall region where $0< y^+ <100$, the wall function is applied. In this region, the Reynolds stresses $\overline{\rho \mathbf{v}'\mathbf{v}'}$ are nearly constant, so is the kinetic energy K. The convection and diffusion in the K-equation are negligible so that local equilibrium prevails. When buoyancy effects are absent, this implies that all source terms in the K-equation are zero. We further recall the definition of the shear stress and obtain

$$\mu_t (v')^2 = \rho\varepsilon, \quad \mu_t v' = \rho v_*^2$$

The solutions of the above equations determine the variables K and ε in the near-wall region

$$K = \frac{v_*^2}{\sqrt{c_\mu}}, \quad \varepsilon = v_*^2 v' \tag{1.10}$$

The wall conditions are determined at a distance, y_w, near the wall. Although the location is not the same as the location of the physical wall, we use the nodal position to store and then plot the solutions that are actually located at y_w. Applying the wall function to Eq.(1.10), we have a complete set of conditions for the fluid flow with turbulence

$$\mathbf{v}_w - \mathbf{w}_w = v_* W(y_w^+)\boldsymbol{\tau}, \quad K_w = \frac{v_*^2}{\sqrt{c_\mu}}, \quad \varepsilon_w = \frac{\rho v_*^4}{\mu_0} W'(y_w^+)$$

where $\boldsymbol{\tau}$ is the normalized tangential direction of the wall. The near-wall location y_w is evaluated as

$$y_w^+ = \min\{\bar{d}_w^+, \tfrac{1}{2}\Delta y^+\}$$

where $\bar{d}_w^+$ is an empirical constant (initially set as 70) and Δy^+ is the normalized near-wall element size.

Although similar research has been conducted for temperature conditions (relating the heat flux at the wall to the difference between the wall temperature and the temperature just outside the viscous sublayer), no convincing results have been reported. We therefore have not included any modification of the temperature value in wall conditions.

The near wall treatment of the fluid and the turbulence variables, as discussed above, is automatically enforced whenever a wall condition or its analogue is applied. These conditions are

- No-slip fixed and moving wall conditions
- Moving wall condition (type=tangential)
- Moving wall condition (type=rotational)
- No-slip fluid-structure interface
- Fluid-structure interface (type=tangential)
- Fluid-structure interface (type=rotational)

7.4.4 User-supplied boundary condition

These conditions are defined in the same user-supplied subroutines and are applied in the same manner as for incompressible fluid flows and temperature conditions (see Section 3.4). The only difference is that conservative variables are used here. That is, in the case $I_{\text{var}} = 1$, $\mathbf{\Phi}$ is defined as $(\rho u, \rho v, \rho w, \rho, \rho E)$.

7.5 Initial conditions

Recalling that pressure, temperature and density must satisfy the state equation that is represented by the ideal gas law, zero pressure or zero temperature must not be specified. Since the default initial condition to all variables is zero, initial conditions are always required in both transient and steady-state analyses. A good initial condition, as will be seen in Chapter 13, may accelerate the convergence during equilibrium iterations. In certain cases, the initial condition may become a key factor in reaching a converged solution.

Initial conditions can be improved in restart analyses or using the solutions mapped from a similar solution.

7.6 Material models

material model: K-ε turbulence

All the material models that are available for high-speed compressible flows are also applicable to the turbulence model described here. In these material models, the parameters corresponding to fluid flows are the same as described in Section 6.6. Additional empirically determined model constants are required in the turbulence model. They are c_μ, c_1, c_2, c_3, σ_K, σ_ε, σ_θ, d_w^+ and κ_0. Their default values are

$$c_\mu = 0.09, \quad c_1 = 1.44, \quad c_2 = 1.92,$$
$$c_3 = 0.8, \quad \sigma_K = 1, \quad \sigma_\varepsilon = 1.3,$$
$$\sigma_\theta = 0.9, \quad d_w^+ = 70, \quad \kappa_0 = 0.4$$

Chapter 8 Formulation of mass transfer in fluid flows

In this chapter, we introduce our formulation of mass transfer in fluid flows.

The model introduced here can be coupled with all fluid models in ADINA-F, including the models of incompressible, slightly compressible, low and high speed compressible flows and flows in porous media. If these fluid models are coupled with ADINA solid models in fluid-structure interaction analyses, the mass transfer solution can also directly be used.

Solving a mass transfer problem involves the solution of the governing equations associated with the input of physical fluid and species material data, well-posed boundary conditions and initial conditions. The formulation of the fluid models has been introduced in the previous chapters. In this chapter, only the mass transfer equations are discussed.

The following table presents a quick overview of the capabilities available for mass transfer analyses.

Table 8-1 Functionalities for mass transfer analyses

Category	Functionality
Coupled models Note: Heat transfer must be coupled in compressible flow models.	ADINA solid model
	incompressible, slightly compressible and compressible flows (including laminar and turbulence models)
	porous media flows
	solid element groups
	heat transfer
	Electro-static and steady current conduction analyses
Computational domains	2D planar, 2D axisymmetric and 3D
Analyses	steady-state and transient
Galerkin Elements	2D: 3-node linear element and 6/9-node bilinear elements
	3D: 4-node linear element and 27-node bilinear elements
FCBI and FCBI-C elements	2D: 3/4-node linear elements
	3D: 4/5/6/8-node linear elements
Material models	constant
	velocity-dependent
	pressure-temperature-dependent
	user-supplied

Boundary conditions	prescribed mass-ratio
	distributed mass flux
	mass convection
	user-supplied
Initial conditions	zero (default conditions)
	specified
	mapped from other solutions
	restart run
Solvers for linearized equations	Gauss elimination method (COLSOL)
	sparse solver
	iterative methods (RPBCG, RPGMRES, and AMG solvers)
Other capabilities	automatically nondimensional procedure
	automatic time-stepping CFL option
	automatic time-stepping ATS option
	others depending on the coupled fluid models.

8.1 Governing equations

The governing equations for mass-ratios and the bulk density are listed in Table 2-1. In the current approach, we assume that the material data are the same for all species. The solution variables are $\phi_i \left(i = 1, 2, ..., n\right)$. Note that the fluid model must be included since the mass transfer equations are always coupled with the fluid flow equations. In the case of moving mesh problems, the ALE formulation described in Section 2.14 is used.

Mass transfer may be active in the whole computational domain, or may be inactive in certain regions–for example, in solid element groups. The mass-ratio can be deactivated in element groups. Once it is inactive, a zero mass flux condition is assumed along the interface between the active-mass and inactive mass regions.

8.2 Numerical method

The time integration schemes are those used for the fluid flow equations.

The finite element method is used to discretize the governing equations. The finite element equations are obtained by establishing a weak form of the equations using the Galerkin procedure. The governing equations are

weighted with the virtual quantity of mass-ratios and integrated over the computational domain V. The divergence theorem is used to lower the order of the derivatives of the mass flux.

Corresponding to the governing equation, the variational form of the mass transfer equation is

$$\int_V \left(h^{\phi_i} G_i + \mathbf{q}_i \bullet \nabla h^{\phi_i} \right) dV = \oint_S h^{\phi_i} \mathbf{q}_i \bullet d\mathbf{S}$$

where h^{ϕ_i} is the virtual quantity of ϕ_i, and

$$G_i = \frac{\partial \rho \phi_i}{\partial t} + \nabla \bullet \left(\rho \mathbf{v} \phi_i \right) - m_i$$
$$\mathbf{q}_i = \mathbf{D} \bullet \nabla \phi_i$$

The upwinding techniques are used while constructing the finite element equations.

In FCBI 2D 4-node and 3D 8-node elements, a finite volume method is used. Using the related interpolation functions described in Chapter 10, the discretized equations are

$$\int_V \left(\frac{\partial \rho \phi_i}{\partial t} - m_i \right) dV + \oint_S \left[\left(\mathbf{v} - \mathbf{w} \right) \rho \phi_i - \mathbf{D} \bullet \nabla \phi_i \right] \bullet d\mathbf{S} = 0$$

In the following cases, the mass solution variables may be coupled with fluid variables.

- any $\alpha_i \neq 0, \quad 1 \leq i \leq n$
- any $\beta_i \neq 0, \quad 1 \leq i \leq n$
- any fluid material depends on mass-ratios

However, the choice of coupling between the fluid variables and the mass-ratios is determined by the user. In certain applications, a decoupled solution may be required.

The mass transfer equations are solved separately from the fluid variables. However, in the case that they are coupled with fluid, the iterative or partitioned procedure is applied. In the Newton-Rapson

method, the sparse solver is always used to solve the mass transfer equations.

8.3 Elements

The elements that can be used for mass transfer are the same as those for incompressible flows.

8.4 Boundary conditions

The boundary conditions for the fluid variables are the same as for fluid models. The conditions introduced here are only for mass-ratios.

When n species are modeled, n mass transfer equations are solved. For each species, a complete set of boundary conditions must be applied.

A typical set of boundary conditions for one species may include prescribed mass-ratios at the inlet, and zero mass flux on the walls and on the outlet. Also, a mass convection condition may be applied to some open boundaries. A general guide for forming a well-posed species equation is to apply one and only one boundary condition at each part of the boundary, and at least one prescribed mass-ratio or mass convection condition is applied if a steady-state analysis is performed. The available conditions are introduced here.

8.4.1 Prescribed mass-ratio

In this condition, a time-dependent mass-ratio is directly prescribed

$$\phi_i = \overline{\phi}_i(t)$$

and applied to boundaries. The mass transfer equations corresponding to the species i at the boundary nodes are then replaced by this equation.

This condition is usually applied at inlet boundaries.

8.4.2 Distributed mass flux

In this condition, a mass flux $\overline{q}_{in}(t)$ is imposed onto the fluid domain. The applied mass flow is then computed by

$$Q_i(t) = \int h^{\phi_i} \overline{q}_{in}(t)\, dS$$

where h^{ϕ_i} is the virtual quantity of mass-ratio on the boundary. The computed mass flow $Q_i(t)$ is directly applied to the right-hand side of the mass transfer equation of the boundary nodes. In particular, when $\overline{q}_{in}(t) = 0$, the application of the mass flux is trivial because it is equivalent to no mass flux loads.

This condition is usually applied to boundaries where the mass flux is known. It has no effect if applied to a boundary where the mass-ratio has been prescribed.

8.4.3 Mass Convection

In this condition, the mass flux from an external convection condition is specified as

$$q_{in} = \overline{h}_i \left(\overline{\phi}_{ie} - \phi_i \right)$$

where $\overline{h}_i$ is a mass convection coefficient and $\overline{\phi}_{ie}$ is an environmental mass-ratio, both could being constant, time-dependent or mass-ratio dependent. The applied mass flow is then computed by

$$Q_i(t) = \int h^{\phi_i} q_{in}\, dS$$

where h^{ϕ_i} is the virtual quantity of mass-ratio on the boundary. The mass flow $Q_i(t)$ is discretized and assembled accordingly into the right-hand side and the stiffness matrix of the mass transfer equation.

This condition is usually applied to the boundary that is exposed to an open environment. It has no effect if applied to a boundary where a mass-ratio condition is prescribed.

8.4.4 User-supplied boundary condition

This condition is described in Section 3.4.

8.5 Initial conditions

In transient analyses, besides initial conditions for all fluid flow variables, all mass-ratios must be specified. The default initial conditions are zero. Although mass-ratio initial conditions are not required in steady-state analyses, they are used as starting values in the equilibrium iterations. Good initial conditions, as will be seen in Chapter 13, may accelerate the convergence during equilibrium iterations. In certain cases, the initial conditions may become the key factor in reaching a converged solution.

Initial conditions can be improved in restart analyses or using the solutions mapped from a similar solution.

8.6 Material models

Fluid flow material conditions are defined when the fluid equations are considered. In this section, only the material conditions for the species are described. Mass material data are assumed to be the same for each of the species.

8.6.1 Constant material data model

This model contains the following parameters:

D_0 = mass diffusion coefficient
β_i = mass volume expansion coefficient for *i*-th species
ϕ_{i0} = mass-ratio reference for *i*-th species

The mass diffusion coefficient tensor is computed by

$$\mathbf{D} = D_0 \mathbf{I}$$

where **I** is the diagonal matrix.

8.6.2 Velocity-dependent model

In this model, the diffusion coefficient tensor is defined as

$$\mathbf{D} = D_0 \mathbf{I} + D_n \mathbf{nn} + D_\tau (\mathbf{I} - \mathbf{nn})$$

where $\mathbf{n}$ is the flow direction, D_0 is a diffusion coefficient, D_n is the diffusion coefficient in the flow direction and D_τ is the diffusion coefficient in the transverse direction of the flow.

The following parameters are required:

D_0 = a homogeneous mass diffusion coefficient

D_n = mass diffusion coefficient in flow direction
D_τ = mass diffusion coefficient in transverse direction
β_i = mass volume expansion coefficient for *i*-th species
ϕ_{i0} = mass-ratio reference for *i*-th species

8.6.3 Pressure-temperature-dependent model

In this model, the mass diffusion coefficient matrix is defined as

$$\mathbf{D} = D_0 \left(\frac{p}{p_0} \right)^{n_p} \left(\frac{\theta}{\theta_0} \right)^{n_\theta} \mathbf{I}$$

It is important to use proper units such that the pressure and temperature solutions can never be zero.

The following parameters are required

D_0 = mass diffusion coefficient at (p_0, θ_0)
p_0 = reference pressure
θ_0 = reference temperature
n_p = exponent of the pressure ratio
n_θ = exponent of the temperature ratio
β_i = mass volume expansion coefficient for *i*-th species
ϕ_{i0} = mass-ratio reference for *i*-th species

8.6.4 User-supplied materials

In this model, the user can specify the material data through a user-supplied subroutine. The subroutine is invoked whenever material data is required in the computations. Any necessary data is passed in as arguments. The arguments are the coordinates of the current spatial point, time, the element group label, the element label, current solution variables and their derivatives at the point, etc.

In return, the following parameters need to be computed or specified in the subroutine

$\mathbf{D}$	= mass diffusion coefficient matrix
ρ	= bulk density
S_0, S_f	= coefficients of the mass creation rate
β_M	= mass buoyancy force ratio

where β_M is defined as

$$\beta_M = \sum_{i=1}^{n} \beta_i (\phi_i - \phi_{i0})$$

and S_0 and S_f are used to compute the mass creation rate according to

$$m_i = S_0 - S_f \phi_i$$

The reason that the program requires the coefficients S_0 and S_f rather than the mass creation rate m_i itself is to assemble the mass-ratio in an implicit way, therefore maintaining the stability of the computational procedure.

The following sample example demonstrates an application of a user-supplied material set for mass transfer equations. In this problem, two species are assumed.

For species-1:

$$\rho = 1, \quad S_0 = S_f = 0, \quad \beta_M = 0$$

$$\mathbf{D} = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 0.2 & 0 \\ 0 & 0 & 0.2 \end{bmatrix}$$

and for species-2

$$\rho = 1, \quad S_0 = S_f = 0, \quad \beta_M = 0$$

$$\mathbf{D} = \begin{bmatrix} 0.2 & 0 & 0 \\ 0 & 1.2 & 0 \\ 0 & 0 & 0.2 \end{bmatrix}$$

In the commented lines, a velocity-dependent material is used. Furthermore, the buoyancy term is defined. These conditions correspond to the following data set

$$\rho = 118.6, \quad S_0 = S_f = 0, \quad \beta_M = 0.1\phi_1$$
$$\mathbf{D} = 0.01\mathbf{I} + 0.01\mathbf{nn} + 0.003(\mathbf{I} - \mathbf{nn})$$

```
      SUBROUTINE ADFURM(MODE,TIME,IELG,IELM,IMAS,NNOD,NM,H,DH,VESL,
     .                  DEN,DIF,S0,SF,BM)
C=======================================================================
C
C     USER SUPPLIED MATERIAL DATA AND SOURCE TERMS FOR MASS TRANSFER
C
C-----------------------------------------------------------------------
C
C     INPUT:
C
C        MODE         = REFERRING PHASE (SEE EXPLANATIONS BELOW)
C        TIME         = CURRENT TIME
C        IELG         = ELEMENT GROUP NUMBER
C        IELM         = ELEMENT NUMBER
C        IMAS         = INDEX OF THE CURRENT SPECIES
C        NNOD         = NUMBER OF LOCAL NODES FOR NNOD-ELEMENT
C        H(*)         = INTERPOLATION FUNCTION
C        DH(3,*)      = DERIVATIVES OF H WITH RESPECT TO (X,Y,Z)
C        VESL( 1,  I) = X-VELOCITY     AT LOCAL NODE I
C        VESL( 2,  I) = Y-VELOCITY     AT LOCAL NODE I
C        VESL( 3,  I) = Z-VELOCITY     AT LOCAL NODE I
C        VESL( 4,  I) = PRESSURE       AT LOCAL NODE I
C        VESL( 5,  I) = TEMPERATURE    AT LOCAL NODE I
C        VESL( 6,  I) = X-DISPLACEMENT AT LOCAL NODE I
C        VESL( 7,  I) = Y-DISPLACEMENT AT LOCAL NODE I
C        VESL( 8,  I) = Z-DISPLACEMENT AT LOCAL NODE I
C        VESL( 9,  I) = TURBULENCE_K   AT LOCAL NODE I
C        VESL(10,  I) = TURBULENCE_E   AT LOCAL NODE I
C        VESL(11,  I) = X-COORDINATE   AT LOCAL NODE I
C        VESL(12,  I) = Y-COORDINATE   AT LOCAL NODE I
C        VESL(13,  I) = Z-COORDINATE   AT LOCAL NODE I
C        VESL(14+K,I) = MASS RATIO OF SPECIES-K (K=0,1,2,...,NMASS)
C        NM           = LENGTH OF THE FIRST ENTRY OF VESL
C
C-----------------------------------------------------------------------
C
C     OUTPUT MATERIAL DATA FOR MODE=1:
C
C        DEN      = BULK DENSITY OF MIXTURE
C                   (FOR INCOMP.FLUIDS, DEFAULT=SPECIFIED)
C                   (FOR COMP.FLUIDS, DEFAULT=COMPUTED USING STATE EQ.)
```

```
C        DIF(3,3) = DIFFUSIVITY TENSOR (MUST BE SPECIFIED HERE)
C
C-----------------------------------------------------------------------
C
C     OUTPUT MASS CREATION RATE FOR MODE=2:
C
C        DEFINITION:
C
C        MASS CREATION RATE = S0 - SF * (MASS RATIO OF SPECIES-IMAS)
C
C        S0 = THE TERM THAT WILL BE PUT ON THE RIGHT-HAND SIDE
C        SF = COEF. THAT WILL BE PUT INTO THE STIFF MATRIX
C        FOR NUMERICAL STABILITY, ALWAYS MANAGE SUCH THAT SF>=0
C
C-----------------------------------------------------------------------
C
C     OUTPUT BOUYANCY FACTOR BM IN MOMENTUM EQUATION FOR MODE=3
C
C        DEFINITION:
C
C        BM = SUM{ BETMI*(SI-SI_REF) }
C
C        SI     = MASS RATIO OF SPECIES-I
C        SI_REF = REFERENCE MASS RATIO OF SPECIES-I
C        BETMI  = MASS VOLUME EXPANSION COEFFICIENT OF THE SPECIES-I
C
C=======================================================================
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION H(NNOD),DH(3,NNOD),VESL(NM,NNOD)
      DIMENSION DIF(3,3)
C-----------------------------------------------------------------------
      BM        =   0.D0
      DH(1,1)   =   DH(1,1)
      H(1)      =   H(1)
      IELG      =   IELG
      IELM      =   IELM
      TIME      =   TIME
      VESL(1,1) =   VESL(1,1)
C-----------------------------------------------------------------------
      IF(MODE.EQ.1) THEN
C       +-------------------------------------------------+
C       |                                                 |
C       |    SPECIFY DEN AND DIF HERE FOR SPECIES-IMAS    |
C       |                                                 |
C       +-------------------------------------------------+
C
C        EXAMPLE FOR CONSTANT MATERIAL
C
         IF(IMAS.EQ.1) THEN
            DEN      = 1.D0
            DIF(1,1) = 10.D0
            DIF(2,2) = 0.2D0
            DIF(3,3) = 0.2D0
            DIF(1,2) = 0.0D0
            DIF(1,3) = 0.0D0
            DIF(2,1) = 0.0D0
            DIF(2,3) = 0.0D0
            DIF(3,1) = 0.0D0
            DIF(3,2) = 0.0D0
         ELSE IF(IMAS.EQ.2) THEN
            DEN      = 1.D0
            DIF(1,1) = 0.2D0
            DIF(2,2) = 1.2D0
```

```
            DIF(3,3) = 0.2D0
            DIF(1,2) = 0.0D0
            DIF(1,3) = 0.0D0
            DIF(2,1) = 0.0D0
            DIF(2,3) = 0.0D0
            DIF(3,1) = 0.0D0
            DIF(3,2) = 0.0D0
          ENDIF
C
C         EXAMPLE FOR V-DEPENDENT MATERIAL
C
c         IU       =  1
c         IV       =  2
c         IW       =  3
c         EPS      =  1.D-20
C
c         DEN      =  118.6D0
c         DIF0     =  0.01D0
c         DIFVN    =  0.01D0
c         DIFVT    =  0.003D0
C
c         U        =  0.D0
c         V        =  0.D0
c         W        =  0.D0
c         DO 2 I   =  1, NNOD
c         U        =  U + VESL(IU,I)*H(I)
c         V        =  V + VESL(IV,I)*H(I)
c         W        =  W + VESL(IW,I)*H(I)
c2        CONTINUE
c         F0FT     =  DIF0  + DIFVT
c         FNFT     =  DIFVN - DIFVT
c         U2       =  U*U + V*V + W*W
c         U2       =  MAX(U2,EPS)
c         UI2      =  FNFT/U2
c         UU       =  U*UI2
c         VV       =  V*UI2
c         WW       =  W*UI2
c         DIF(1,1) =  U*UU + F0FT
c         DIF(2,2) =  V*VV + F0FT
c         DIF(3,3) =  W*WW + F0FT
c         DIF(1,2) =  U*VV
c         DIF(1,3) =  U*WW
c         DIF(2,3) =  V*WW
c         DIF(3,1) =  DIF(1,3)
c         DIF(3,2) =  DIF(2,3)
c         DIF(2,1) =  DIF(1,2)
          GO TO 9999
        ENDIF
C
        IF(MODE.EQ.2) THEN
C         +-------------------------------------------------+
C         |                                                 |
C         |    SPECIFY THE SOURCE IN TERMS OF S0 AND SF     |
C         |                                                 |
C         +-------------------------------------------------+
           S0  = 0.D0
           SF  = 0.D0
           GO TO 9999
        ENDIF
C
c       IF(MODE.EQ.3) THEN
C         +-------------------------------------------------+
C         |                                                 |
```

```
C       |    SPECIFY BM = SUM{ BETM*(SI-SIREF) } HERE     |
C       |                                                 |
C       +-------------------------------------------------+
c        NMASS   = 2
c        BM      = 0.D0
c        DO 31 I = 1, NMASS
c        IM      = 14 + I
c        SI      = 0.D0
c        SIREF   = 0.D0
c        BETMI   = 0.1D0
c        IF(I.EQ.2) BETMI = 0.D0
c        DO 32 J = 1, NNOD
c        SI      = SI + H(J)*VESL(IM,J)
c32      CONTINUE
c        BM      = BM + BETMI*(SI-SIREF)
c31      CONTINUE
c        GO TO 9999
c     ENDIF
C-----------------------------------------------------------------------
 9999 CONTINUE
      RETURN
      END
```

This page intentionally left blank

Chapter 9 Formulation of fluid-structure interactions

9.1 Introduction

In this chapter, we introduce the formulation used for fluid-structure interactions (FSI) in the ADINA system. The fluid flow models used here can be incompressible flows, slightly compressible flows, low and high speed compressible flows and flows through porous media. The solid models used here can be all those available in ADINA. Heat and mass transfers can also be specified in fluid-structure interaction analyses.

In fluid-structure interaction analyses, fluid forces are applied onto the solid and the solid deformation changes the fluid domain. For most interaction problems, the computational domain is divided into the fluid domain and solid domain, where a fluid model and a solid model are defined respectively, through their material data, boundary conditions, etc. The interaction occurs along the interface of the two domains. Having the two models coupled, we can perform simulations and predictions of many physical phenomena.

The ADINA system has a structural analysis capability as well as a fluid analysis capability. The availability of both capabilities within the same code provides the base for developing sophisticated fluid-structure interaction tools.

A typical fluid-structure interaction problem is illustrated in the next figure. The fluid model is defined in the fluid domain with wall boundary conditions, prescribed velocity at the inlet, zero distributed normal-traction at the outlet and, most importantly, the fluid-structure interface condition. The solid model is defined in the structural domain, where its bottom is fixed and its top is the fluid-structure interface. The two models must have also been discretized using the elements that are available in ADINA Solids & Structures and ADINA-F, respectively. The typical task of an analysis of a fluid-structure model is to obtain the fluid and structure response through the coupled solution. The structural model is based on a Lagrangian coordinate system and the displacements are the primary

unknowns. A pure fluid model is always analyzed using a Eulerian coordinate system. However, for fluid-structure interaction problems, the fluid model must be based on an arbitrary-Lagrangian-Eulerian coordinate system since the fluid-structure interface is deformable. Therefore, the solution variables of the fluid flow include the usual fluid variables (pressure, velocity, etc.) as well as displacements.

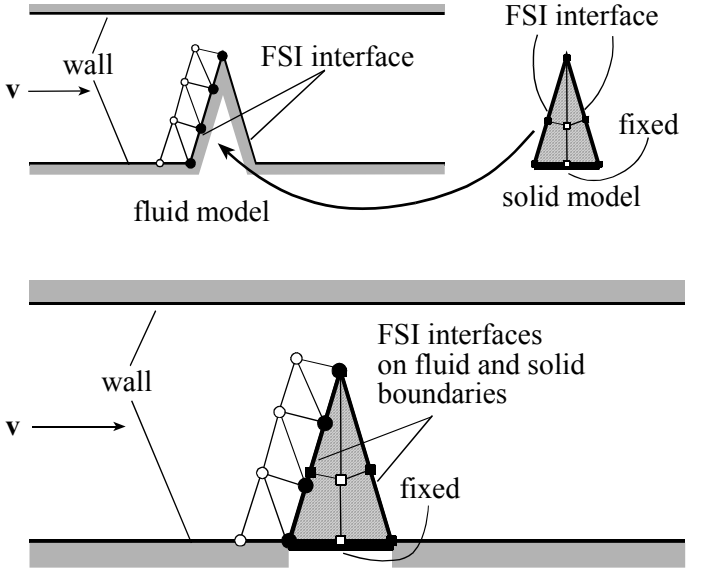


Figure 9.1 Illustration of fluid-structure interactions

9.2 Theory

ref. Zhang, H., and Bathe, K.J., "Direct and Iterative Computing of Fluid Flows fully Coupled with Structures," *Computational Fluid and Solid Mechanics,* K.J. Bathe, editor, Elsevier Science, 2001.

9.2.1 Kinematic and dynamic conditions

The fundamental conditions applied to the fluid-structure interfaces are the kinematic condition (or displacement compatibility)

$$\underline{\mathbf{d}}_f = \underline{\mathbf{d}}_s$$

and the dynamic condition (or traction equilibrium)

$$\mathbf{n} \bullet \underline{\boldsymbol{\tau}}_f = \mathbf{n} \bullet \underline{\boldsymbol{\tau}}_s$$

where $\underline{\mathbf{d}}_f$ and $\underline{\mathbf{d}}_s$ are, respectively, the fluid and solid displacements and $\underline{\boldsymbol{\tau}}_f$ and $\underline{\boldsymbol{\tau}}_s$ are, respectively, the fluid and solid stresses. The underlining denotes that the values are defined on the fluid-structure interfaces only. The fluid velocity condition is resulted from the kinematic condition

$$\underline{\mathbf{v}} = \dot{\underline{\mathbf{d}}}_s$$

if a no-slip condition is applied, or

$$\mathbf{n} \bullet \underline{\mathbf{v}} = \mathbf{n} \bullet \dot{\underline{\mathbf{d}}}_s$$

if a slip condition is applied.

The fluid and solid models are coupled as follows.

The fluid nodal positions on the fluid-structure interfaces are determined by the kinematic conditions. The displacements of the other fluid nodes are determined automatically by the program to preserve the initial mesh quality. The governing equations of fluid flow in their ALE formulations are then solved. In steady-state analyses, the mesh velocities are always set to zero even the fluid nodal displacements are updated. Accordingly, the fluid velocities on the fluid-structure interfaces are zero.

According to the dynamic conditions, on the other hand, the fluid traction is integrated into fluid force along fluid-structure interfaces and exerted onto the structure node

$$\underline{\mathbf{F}}(t) = \int h^{\mathrm{d}} \underline{\boldsymbol{\tau}}_f \bullet d\mathbf{S} \qquad (1.11)$$

where $h^{\mathbf{d}}$ is the virtual quantity of the solid displacement.

9.2.2 Separate meshes of fluid and solid models

Completely different elements and meshes can be used in fluid and solid models. The elements are solely limited by the availability of solid and fluid models in the ADINA system. The nodal point positions of the two models are therefore generally not the same on the fluid-structure interface as illustrated in Fig. 9.1 and in the following figure.

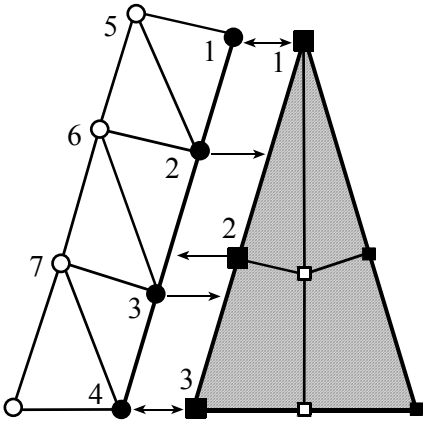


Figure 9.2 Coupling of fluid and solid nodes

The fluid nodal displacements are interpolated using the solid nodal displacements. For example, the displacement at fluid node 2 is interpolated using the displacements at solid nodes 1 and 2. Given the boundary nodal displacements, the other fluid nodal displacements are calculated in a special procedure to preserve the initial mesh quality. The procedure is generally applicable to all types of moving meshes and thus is described in detail in Section 12.10.

Similarly, the fluid traction at a solid node is interpolated using the stress of the fluid boundary element where the solid node is located. In the example shown here, the fluid stress at solid node 2 is interpolated using the fluid stresses at fluid nodes 2 and 3, while the stresses at solid nodes 1

and 3 equal the fluid stresses at fluid nodes 1 and 4 respectively. According to Eq.(1.11), the fluid force at solid node 2 includes stress at solid nodes 1, 2 and 3. Recall that the fluid stress is the sum of the pressure and the shear stress. The fluid solution variables and the solid solution variables at the numbered nodes are therefore eventually all coupled.

Since we allow separate meshes in the fluid and solid domains, it is likely that the two meshes on the interfaces are not compatible. The discretized representations of the same geometry in the two models may therefore not be the same. However, the distance between the two discretized boundaries must be within a small distance. We define a relative distance from fluid FSI nodes to the solid FSI interface as

$$r_f = \max\left\{\frac{d_f}{D_s}\right\}$$

where d_f is the distance from a fluid node to the structural discretized boundary and D_s is the length of the solid boundary element (see the figure below). This relative distance is always checked. Whenever $r_f \geq 1$, the program sees this as an error and stops. Whenever $0.001 \leq r_f < 1$, a warning message is printed and the program continues.

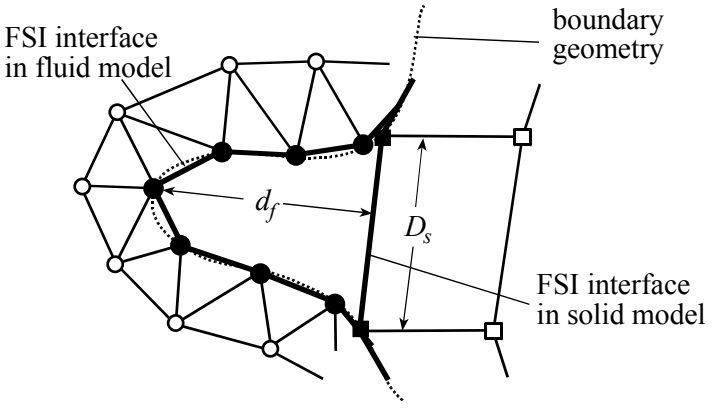


Figure 9.3 Measure of the distance between fluid and solid FSI boundaries

If there are gaps between fluid FSI nodes and the solid FSI interface, these fluid nodes are automatically moved onto the solid interface and the other fluid nodes are moved accordingly in the same procedure as described in Section 12.10.

Similarly a relative distance from the solid FSI nodes to the fluid FSI interface is defined and checked

$$r_s = \max\left\{\frac{d_s}{D_f}\right\}$$

where d_s is the distance from a solid node to the fluid boundary and D_f is the size of the fluid boundary element. According to the value of r_s, the program prints a warning/error message and continues/stops as for fluid nodes.

When the distance between the boundaries is too large, either the geometries used in the fluid and solid models are apart from each other, or the boundary elements have misrepresented the geometry in one or both models.

To avoid possible discrepancies between the two interfaces, the following tips could be useful.

- Unnecessary details of the geometry must be cleaned out prior to generating meshes.
- The mesh should be fine enough to represent the boundary geometry, particularly along locations of sharp curvatures. If fewer elements are intentionally used along the FSI boundary where sharp curvatures exist, or (because of the limitation of the computer capacity) the boundary elements are unable to reasonably represent the geometry in any one of the models, it is best to simplify the geometry.
- Along boundaries of sharp curvatures, use compatible meshes. Note that even when the elements have the same size for the fluid and solid models, the boundary nodes of both models may not be at the same locations. Incompatible meshes can generally be applied to smooth boundaries.

9.2.3 Consistent time integration for fluid and solid models

The time integrations for both fluid and solid equations must be consistent. Although different coordinate systems are used in the fluid and solid models, the two systems are the same on fluid-structure interfaces where the Lagrangian coordinate system is used. We therefore first focus on the time integration on the interface and then apply the results to the whole computational domain. Since the displacement, velocity and acceleration on the interface are the same for both fluid and solid models, we will not distinguish between them on the interfaces.

Let the fluid equations and the solid equations be represented by $\mathbf{G}_f\left[\mathbf{f},\dot{\mathbf{f}}\right]=\mathbf{0}$ and $\mathbf{G}_s\left[\mathbf{d},\dot{\mathbf{d}},\ddot{\mathbf{d}}\right]=\mathbf{0}$ respectively, where the fluid variables are represented by $\mathbf{f}$ and the solid displacements are represented by $\mathbf{d}$. We also consider that the fluid flow equations are balanced at time $t+\alpha\Delta t$ and the solid equations are satisfied at time $t+\Delta t$. Consider the Euler method used for the fluid velocity and acceleration

$$
\begin{aligned}
{}^{t+\alpha\Delta t}\underline{\mathbf{v}} &\equiv \frac{{}^{t+\Delta t}\underline{\mathbf{d}}-{}^{t}\underline{\mathbf{d}}}{\Delta t} = {}^{t+\Delta t}\underline{\mathbf{v}}\,\alpha + {}^{t}\underline{\mathbf{v}}(1-\alpha)\\
{}^{t+\alpha\Delta t}\underline{\mathbf{a}} &\equiv \frac{{}^{t+\Delta t}\underline{\mathbf{v}}-{}^{t}\underline{\mathbf{v}}}{\Delta t} = {}^{t+\Delta t}\underline{\mathbf{a}}\,\alpha + {}^{t}\underline{\mathbf{a}}(1-\alpha)
\end{aligned}
$$

From these equations, the velocity and the acceleration at time $t+\Delta t$ can be obtained in terms of the unknown displacement

$$
\begin{aligned}
{}^{t+\Delta t}\underline{\mathbf{v}} &= \frac{1}{\alpha\Delta t}\left({}^{t+\Delta t}\underline{\mathbf{d}}-{}^{t}\underline{\mathbf{d}}\right)-{}^{t}\underline{\mathbf{v}}\left(\frac{1}{\alpha}-1\right) \equiv {}^{t+\Delta t}\underline{\mathbf{d}}\,a+{}^{t}\boldsymbol{\xi}\\
{}^{t+\Delta t}\underline{\mathbf{a}} &= \frac{1}{\alpha^2\Delta t^2}\left({}^{t+\Delta t}\underline{\mathbf{d}}-{}^{t}\underline{\mathbf{d}}\right)-{}^{t}\underline{\mathbf{v}}\frac{1}{\alpha^2\Delta t}-{}^{t}\underline{\mathbf{a}}\left(\frac{1}{\alpha}-1\right) \equiv {}^{t+\Delta t}\underline{\mathbf{d}}\,b+{}^{t}\boldsymbol{\eta}
\end{aligned}
$$

We then apply these equations to the coupled system. The final time integration scheme can therefore be summarized as

$$
\begin{aligned}
{}^{t+\alpha\Delta t}\mathbf{G}_f &\approx \mathbf{G}_f\left[{}^{t+\alpha\Delta t}\mathbf{f},\left({}^{t+\alpha\Delta t}\mathbf{f}-{}^{t}\mathbf{f}\right)/\alpha\Delta t\right]=\mathbf{0}\\
{}^{t+\Delta t}\mathbf{G}_s &\approx \mathbf{G}_s\left[{}^{t+\Delta t}\mathbf{d},\,{}^{t+\Delta t}\mathbf{d}\,a+{}^{t}\boldsymbol{\xi},\,{}^{t+\Delta t}\mathbf{d}\,b+{}^{t}\boldsymbol{\eta}\right]=\mathbf{0}
\end{aligned}
$$

It can be shown that the necessary stability condition of the consistent time integration is $\alpha \geq 1/2$. The condition is obtained based on the assumptions of a linear system and equal size elements. The scheme of $\alpha = 1/2$ also presents a scheme of second order accuracy in time. However, since the coupled system is nonlinear and the elements are practically non-uniform, the scheme of $\alpha = 1/2$ could be unstable or oscillating. For most applications, $\alpha = 1$ is recommended.

The Composite time integration method of second order accuracy (see section 3.2.1 for details) is also available for FSI.

9.2.4 Finite element equations of the coupled system

Let the solution vector of the coupled system be $\mathbf{X} = (\mathbf{X}_f, \mathbf{X}_s)$, where $\mathbf{X}_f$ and $\mathbf{X}_s$ are the fluid and solid solution vectors defined at the fluid and solid nodes respectively. Thus, $\underline{\mathbf{d}}_s = \underline{\mathbf{d}}_s(\mathbf{X}_s)$ and $\underline{\boldsymbol{\tau}}_f = \underline{\boldsymbol{\tau}}_f(\mathbf{X}_f)$. The finite element equations of the coupled fluid-structure system can be expressed as

$$\mathbf{F}[\mathbf{X}] \equiv \begin{bmatrix} \mathbf{F}_f\left[\mathbf{X}_f, \underline{\mathbf{d}}_s(\mathbf{X}_s)\right] \\ \mathbf{F}_s\left[\mathbf{X}_s, \underline{\boldsymbol{\tau}}_f(\mathbf{X}_f)\right] \end{bmatrix} = \mathbf{0} \qquad (1.12)$$

in which, $\mathbf{F}_f$ and $\mathbf{F}_s$ are finite element equations corresponding to equations $\mathbf{G}_f$ and $\mathbf{G}_s$ respectively. Note that the decoupled fluid and solid equations can be represented by $\mathbf{F}_f\left[\mathbf{X}_f, \mathbf{0}\right] = \mathbf{0}$ and $\mathbf{F}_s\left[\mathbf{X}_s, \mathbf{0}\right] = \mathbf{0}$ respectively.

- **Two-way fluid-structure coupling**

For many coupled problems, the fluid traction affects the structural deformations and the solid displacement affects the flow pattern. This fact is the reason for performing fluid-structure interaction analyses. We call this type of analysis “two-way coupling”.

The coupled fluid-structure equation is a nonlinear system regardless of the solid model used (linear or nonlinear), since the fluid equations are always nonlinear. An iteration procedure must therefore be used to obtain the solution at a specific time. In other words, we have iterative solutions $\mathbf{X}^1, \mathbf{X}^2, \ldots$ in fluid-structure interaction problems. We use criteria that are based on either stress or displacement, or both of these, to check for convergence of the iterations. The stress criterion is defined as

$$r_\tau \equiv \frac{\left\| \underline{\boldsymbol{\tau}}_f^k - \underline{\boldsymbol{\tau}}_f^{k-1} \right\|}{\max\left\{ \left\| \underline{\boldsymbol{\tau}}_f^k \right\|, \varepsilon_0 \right\}} \leq \varepsilon_\tau$$

and the displacement criterion is defined as

$$r_d \equiv \frac{\left\| \underline{\mathbf{d}}_s^k - \underline{\mathbf{d}}_s^{k-1} \right\|}{\max\left\{ \left\| \underline{\mathbf{d}}_s^k \right\|, \varepsilon_0 \right\}} \leq \varepsilon_d$$

where ε_τ and ε_d are tolerances for stress and displacement convergence, respectively, and ε_0 is a pre-determined constant ($\equiv 10^{-8}$) for the purpose overriding of stresses or displacements in case they become too small to measure convergence. The default option is that both criteria need to be satisfied.

- **One-way fluid-structure coupling**

In certain cases, the deformation of the solid is so small that its influence on the fluid flow is negligible. Then only the fluid stress needs to be applied onto the structure and no iteration between the fluid and solid models is needed. We call this type of interaction "one-way coupling".

To reach solution effectiveness, we can either directly or iteratively solve the coupled system. In one-way coupling, we can also solve the system together in one run or separately in a number of runs. All these options are discussed in the next sections.

In all cases, the fluid and solid solutions are saved in different porthole files, namely <*fluid*>.por and <*solid*>.por.

9.3 Iterative computing of two-way coupling

This computing method is also called the partitioned method. In this solution, the fluid and solid solution variables are fully coupled. The fluid equations and the solid equations are solved individually in succession, always using the latest information provided from another part of the coupled system. This iteration is continued until convergence in the solution of the coupled equations is reached.

The computational procedure can be summarized as follows. To obtain the solution at time $t + \Delta t$, we iterate between the fluid model and the solid model. We start with the initial solution guess $\underline{\mathbf{d}}_s^{-1} = \underline{\mathbf{d}}_s^0 = {}^t\underline{\mathbf{d}}_s$ and $\underline{\boldsymbol{\tau}}_f^0 = {}^t\underline{\boldsymbol{\tau}}_f$. For iterations $k = 1, 2, \ldots$, the following equilibrium iteration is performed to obtain the solution ${}^{t+\Delta t}\mathbf{X}$.

1) Solve the fluid solution vector $\mathbf{X}_f^k$ from the fluid equation

$$\mathbf{F}_f \left[\mathbf{X}_f^k, \lambda_d \underline{\mathbf{d}}_s^{k-1} + (1 - \lambda_d) \underline{\mathbf{d}}_s^{k-2} \right] = \mathbf{0}$$

This solution is obtained in fluid flow analysis, using the prescribed solid displacement. Note that the solid displacement has been relaxed by using a displacement relaxation factor λ_d $(0 < \lambda_d \leq 1)$, which may be useful for many difficult problems since the fluid and solid models are not solved in the same matrix system. The use of this factor helps to reach convergence in the iterations.

2) If only the stress criterion needs to be satisfied, the stress residual is computed and checked against the tolerance. If the criterion is satisfied, steps 3 to 5 are skipped.
3) Solve the solid solution vector $\mathbf{X}_s^k$ from the structural equation

$$\mathbf{F}_s \left[\mathbf{X}_s^k, \lambda_\tau \underline{\boldsymbol{\tau}}_f^k + (1 - \lambda_\tau) \underline{\boldsymbol{\tau}}_f^{k-1} \right] = \mathbf{0}$$

Note that the fluid stresses have also been relaxed using a stress relaxation factor λ_τ $(0 < \lambda_\tau \leq 1)$.

4) The fluid nodal displacements are computed with the prescribed boundary conditions $\underline{\mathbf{d}}_f^k = \lambda_d \underline{\mathbf{d}}_s^k + (1-\lambda_d)\underline{\mathbf{d}}_s^{k-1}$.
5) If only the displacement criterion needs to be satisfied, the displacement residual is computed and checked against the tolerance. If both the stress and the displacement criteria need to be passed, both convergence conditions are checked. If the iteration has not converged yet, the program goes back to step 1 and continues for the next iteration unless a maximum number of FSI iterations have been reached (in this case, the program stops and prints out the divergence information).
6) Print and save the fluid and solid solutions if required.

In this solution method, the time steps and solution times are controlled in the fluid model. However, all time functions defined in the solid model must cover the time range of the computation. The parameters that control the convergence of the coupled system are also determined in the fluid model. These parameters are the stress and displacement tolerances and relaxation factors, convergence criteria (see above), etc.

The convergences reached in fluid and solid equations are controlled in the fluid and solid models respectively. The controls of saving and printing solutions are also determined in the individual models.

The iterative method requires less memory than the direct method in two-way couplings. It is suitable for small to large problems that have or have not contact conditions. It is also good for steady-state analyses.

9.4 Direct computing of two-way coupling

This computing method is also called the simultaneous solution method. In this direct solution method, as in the above iterative solution, the fluid and solid solution variables are also fully coupled. The fluid equations and the solid equations are combined and treated in one system. Therefore, they are linearized in a matrix system, as for a fluid model or a solid model alone. This matrix system can be written as

$$\begin{bmatrix} \mathbf{A}_{ff} & \mathbf{A}_{fs} \\ \mathbf{A}_{sf} & \mathbf{A}_{ss} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{X}_f^k \\ \Delta\mathbf{X}_s^k \end{bmatrix} = \begin{bmatrix} \mathbf{B}_f \\ \mathbf{B}_s \end{bmatrix}$$

and

$$\mathbf{X}^{k+1} = \mathbf{X}^k + \Delta\mathbf{X}^k$$

For example, when the Newton-Raphson method is used, we have

$$\mathbf{B}_f = -\mathbf{F}_f^k \equiv -\mathbf{F}_f\left[\mathbf{X}_f^k, \lambda_d \underline{\mathbf{d}}_s^k + (1-\lambda_d)\underline{\mathbf{d}}_s^{k-1}\right]$$
$$\mathbf{B}_s = -\mathbf{F}_s^k \equiv -\mathbf{F}_s\left[\mathbf{X}_s^k, \lambda_\tau \underline{\boldsymbol{\tau}}_f^k + (1-\lambda_\tau)\underline{\boldsymbol{\tau}}_f^{k-1}\right]$$
$$\mathbf{A}_{ij} = \frac{\partial \mathbf{F}_i^k}{\partial \mathbf{X}_j} \quad (i, j = f, s)$$

Note that the fluid stress and solid displacement have also been relaxed, as in the iterative coupling method, using a stress relaxation factor λ_τ and a displacement relaxation factor λ_d respectively. These factors have been practically proven useful for many difficult problems even in the direct computing method.

The computational procedure can be summarized as follows. We start with the initial solution guess $\mathbf{X}^0 = {}^t\mathbf{X}$ for the solution at time $t + \Delta t$. For iterations k = 1,2,…, the following equilibrium iteration procedure is performed to obtain the solutions ${}^{t+\Delta t}\mathbf{X}$.

(1) Assemble the fluid and solid equations as is normally done in a single fluid and a single solid model, followed by assembling the coupling matrices $\mathbf{A}_{fs}$ and $\mathbf{A}_{sf}$.
(2) Solve the linearized equation of the coupled system and update the solution (including solving the fluid displacements). Compute and check the stress and/or displacement residuals against the specified tolerances. If the solution has not converged yet, the program goes back to step 1 to continue for the next iteration unless a maximum number of FSI iterations have been reached (in this case, the program stops and prints out the divergence information).
(3) Print and save the fluid and solid solutions if required.

In the direct computing method, as in the iterative solution method, the time steps are controlled by the fluid flow model. Hence the same control parameters specified in the solid solver are ignored in the solution procedure. However, all time functions defined in the solid model must cover the time range of the computation. The parameters that control the convergence of the coupled system are also processed in the fluid model. These parameters are the stress and displacement tolerances, relaxation factors, convergence criteria, etc.

The controls of saving and printing solutions are determined in the individual solid and fluid models.

The direct method in general is faster than the iterative method in two-way couplings. It is suitable for small to medium problems that have not contact conditions. It is also good for transient analyses.

For obvious reasons, the direct FSI method is not applicable to the segregated method, where even fluid variables are not directly coupled.

9.5 Direct computing of one-way coupling

In this case, the fluid stress is applied onto the structure while the structure has no influence on the flow field. Choose one as the maximum number of FSI iterations permitted in the iterative fluid-structure coupling for this type of analyses. The program will then perform the following steps in each time step:

1) Solve the fluid solution vector $\mathbf{X}_f$ from the fluid equation $\mathbf{F}_f\left[\mathbf{X}_f, \mathbf{0}\right] = \mathbf{0}$ as for a fluid flow model alone with zero solid displacements.
2) Solve the solid solution vector $\mathbf{X}_s$ from the structural equation $\mathbf{F}_s\left[\mathbf{X}_s, \underline{\boldsymbol{\tau}}_f\right] = \mathbf{0}$ as for a solid model alone with the prescribed fluid stress.
3) Print and save the fluid and solid solutions if required.

In the direct computing of one-way coupling problems, the time step control is specified in the fluid model. The controls of saving and printing solutions are determined in the individual solid and fluid models.

The direct method in one-way couplings is suitable for small to medium problems that have negligibly small displacements to fluid fields.

9.6 Indirect computing of one-way coupling

When the fluid model is large in one-way FSI couplings, it can be effective to first compute the fluid flows and save the stresses in a file at selected time steps. The solid models can then be analyzed, with the fluid stress input from the file, as many times as required. It is important to mention here that the fluid and solid models are still prepared separately, so that the meshes of the two models may not be compatible on the interface. In this indirect computing method, the program performs the same functions as in the direct computing of the one-way coupling.

In the indirect computing of one-way coupling problems, all the control parameters are specified in the individual fluid and solid solvers. There are therefore possible discrepancies between the fluid and solid solution times. At certain times when the fluid stresses are not available, a linear interpolation is performed to provide the fluid stress for the solid model (see the figure below). If the solution time is out of range of the times for which the fluid stresses are saved, a linear extrapolation is applied. In order to have more accurate solutions, the fluid stresses should be saved more frequently to cover the time steps that may be used for the solid model.

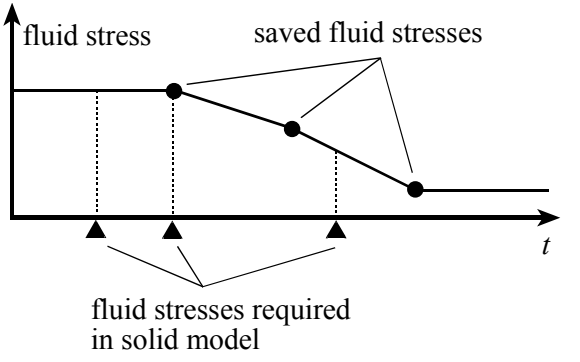


Figure 9.4 Interpolation of fluid stresses in time in indirect

computing of one-way fluid-structure coupling

The indirect method in one-way couplings is suitable for small to large problems that have negligibly small displacements to fluid fields.

9.7 Elements on fluid and solid interfaces

Any fluid element that can be used in a fluid model can also be used in fluid-structure interaction models. These elements are either two-dimensional elements (including axisymmetric elements) or three-dimensional elements. The fluid-structure interfaces are always assigned to boundaries of elements. In two dimensions and three dimensions, the interfaces are then formed by line elements and surface elements, respectively, which are generated automatically by the program.

Any solid element, on the other hand, that can be used in a separate solid model can also be used in fluid-structure interaction models. However, some of the elements cannot be directly connected with fluid elements, which means that no fluid-structure interface can be assigned to those elements. For example, the spring element cannot be directly connected with fluid elements.

As a special case of directly connectable solid elements, the beam-type elements can interact through both their sides with 2D-fluid boundary elements and the shell-type elements can interact through both their surfaces with 3D-fluid boundary elements. We call them F-S-F connections. In this case, the two fluid boundaries are geometrically identical but connected with the two sides/surfaces of the solid elements respectively. On the contrary, when solid boundary elements or one side/surface of the solid elements are connected with fluid boundary elements, we call them F-S connections.

The possible connections of fluid and solid elements are listed in the following table, in which the abbreviation “n.c.” stands for “not connectable”.

Table 9-1 Elements in connections of fluid and solid models in fluid-structure interaction problems

solid elements ⇓	**fluid elements ⇒**	2D fluid elements	3D fluid elements
	connection type ⇒		
truss/cable elements beam elements iso-beam elements pipe elements		F-S F-S-F	n.c.
2D-solid elements plate elements shell elements		F-S	F-S F-S-F
3D-solid elements		n.c.	F-S
2D-displacement-based fluid elements 2D-potential-based fluid elements		F-S	n.c.
3D-displacement-based fluid elements 3D-potential-based fluid elements		n.c.	F-S
spring/damper/mass elements		n.c.	

In the case of F-S-F connections, although the two fluid boundaries are geometrically identical, they are separated by the solid elements and therefore generally nonidentical to the physical quantities, such as the stresses and the pressures. The fluid elements must be carefully modeled in this case and are discussed hereafter.

- **O-shape 2D F-S-F connection**

The following figure shows a typical example of an O-shape two-dimensional fluid element layout, where a beam is surrounded by fluid elements and may be supported by springs that are invisible to fluids. Two fluid boundaries and the solid beam locate at the same place. Since the fluid variables on both sides of the beam are different, the fluid boundary nodes must be different except, however, the two corner nodes that must be coincident since the fluid variables are the same there.

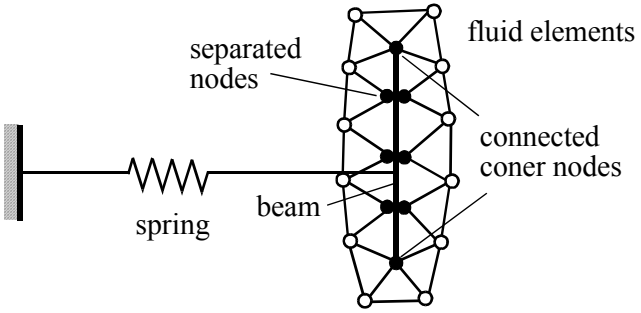


Figure 9.5 Typical example of O-shape 2D F-S-F connections in fluid-structure interaction analyses.

The O-shape two-dimensional fluid elements can be generated in the AUI by explicitly forcing the corner nodes to be coincident while the middle nodes are not checked in the coincident procedure. However, this may be a tedious input if the geometry is complicated (this is particularly true in three-dimensions). Two simple solutions are presented in the following figure.

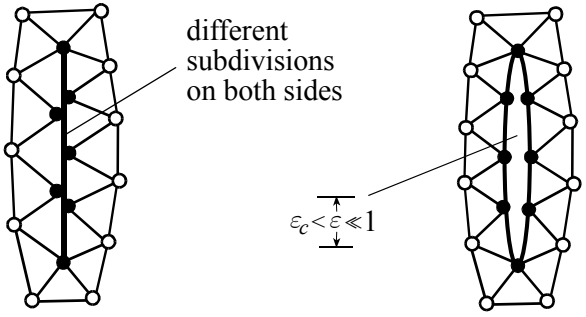


Figure 9.6 Possible methods for generating O-shape 2D fluid elements in F-S-F connections

In the first solution, different element subdivisions are used on the two boundaries. The key here is to make sure that there are no middle nodes at the same locations.

The second solution is to move the two faces slightly apart, while allowing the end-points to still share the same locations. The distance between the two faces must be small enough (for example, less than ten percent of the boundary element length), but must be larger than the tolerance ε_c that is used in the coincidence checking when the elements are generated.

Two fluid-structure interfaces must be defined in the solid model and in the fluid model. Each of them corresponds to one side of the beam. In this example, the fluid-structure interface 1 in the fluid model is connected logically with the fluid-structure interface 1 that is defined in the solid model. They form a pair to model the left side of the beam. Similarly, the pair of the interfaces 2 models the right side of the beam.

In general, the connectable boundaries should be specified in the same fluid-structure interface, since the connected fluid corner nodes may be over specified by different moving conditions. However, the corner nodes in the example are uniquely defined in the two FSI conditions.

Truss/cable elements, iso-beam elements and pipe elements are similar to beam elements in defining fluid-structure interaction problems.

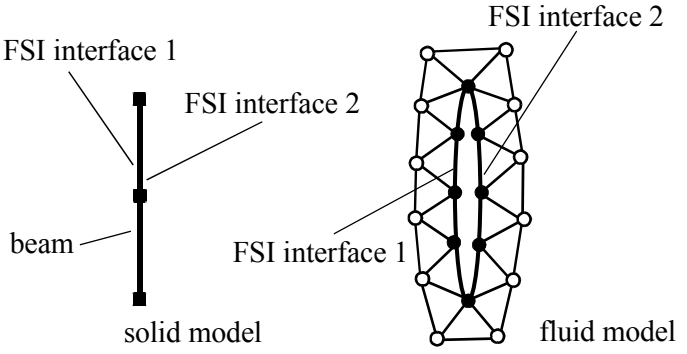


Figure 9.7 Application of fluid-structure interface conditions to beam elements

- **C-shape 2D F-S-F connection**

The following figure shows a typical example of C-shape two-dimensional fluid elements, where a beam is surrounded by fluid and has its one end-point fixed. Two fluid boundaries share the same geometric locations with the solid beam. Except at the fixed end-point, the fluid pressures on both sides of the beam are different. The fluid boundary nodes, therefore, must be separated on the two faces except that the end-node points must be kept coincident.

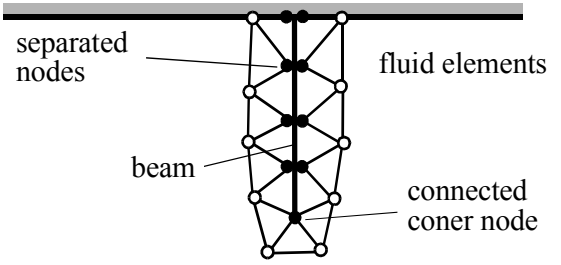


Figure 9.8 Typical example of C-shape 2D F-S-F connections in fluid-structure interaction analyses.

The C-shape two-dimensional fluid elements can be generated in the AUI by explicitly forcing the connected corner node to be coincident while the other nodes are not checked in the coincident procedure. However, this input may be tedious if the geometry is complicated. Two simple solutions are presented in the following figure.

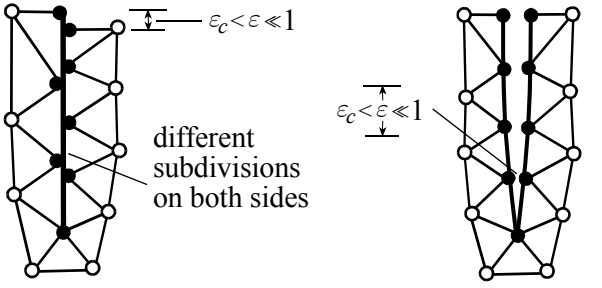


Figure 9.9 Possible methods for generating C-shape 2D fluid elements in F-S-F connections

The first solution is to move the end-point a small distance apart and to use different subdivisions on the two boundaries. The key here is to make sure there are no nodes at the same locations.

The second solution is to move the two faces slightly apart, while allowing only the end-point (the connected one) to remain at the same location. The distance between the two faces must be small enough (for example, less than ten percent of the boundary element length), but must be larger than the tolerance ε_c that is used in the coincidence checking.

As in the case of the O-shape interface, two interfaces must be defined in the solid model and in the fluid model, each pair of the fluid-solid interfaces corresponding to one side of the beam.

Truss/cable elements, iso-beam elements and pipe elements are similar to beam elements in defining fluid-structure interaction problems.

- **O-shape 3D F-S-F connection**

In three dimensions, O-shape fluid elements enclose a plate or a shell. The model needs to be constructed as for the O-shape 2D fluid-structure interface. The solid shell is treated as two faces that are connected with two fluid faces. The elements generated on the two fluid faces must be separated inside the face, while connected on the boundary of the faces. A fluid-structure interface must be assigned to one fluid face and connected to its solid counterpart to form a pair of fluid-structure interfaces. A separate pair is similarly assigned for another face.

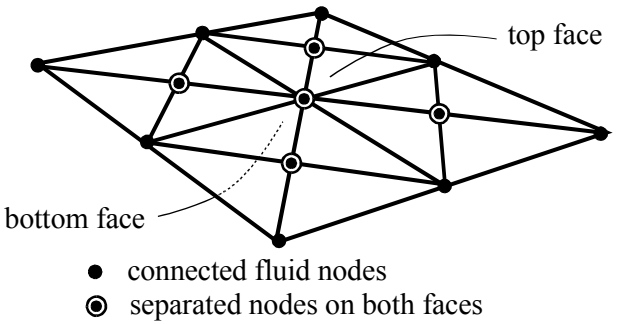


Figure 9.10 Typical example of O-shape 3D F-S-F connections in fluid-structure interaction analyses.

The fluid elements can be generated by enforcing coincidence checking on the boundary of the two fluid faces, while not checking inside the faces. However, other methods that are analogous to those in the two-dimensional case can also be used. For example, by slightly changing the shape of one face, the two faces would be moved apart by a small distance while keeping their boundaries together. The distance between the two faces must be small enough (for example, less than ten percent of the boundary element length), but must be larger than the tolerance used in coincidence checking.

- **C-shape 3D F-S-F connection**

Unlike the O-shape three-dimensional fluid elements, the C-shape fluid elements leave an open part where the shell could be fixed or connected to another part of the model. Along the open part, the fluid nodes must also be separated. The solid shell is treated as two faces that are connected with two fluid faces. The elements generated on the two fluid faces must be separated inside the face and on the open face boundary, while connected on the other boundaries of the two faces. A fluid-structure interface must be assigned to one fluid face and connected to its solid counterpart to form a pair of fluid-structure interfaces. A separate pair is similarly assigned for another face.

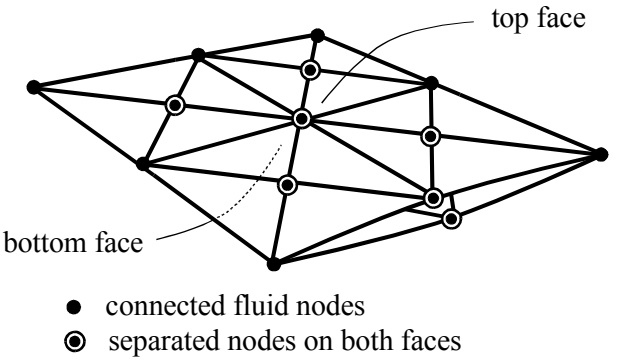


Figure 9.11 Typical example of C-shape 3D F-S-F connections in fluid-structure interaction analyses.

The fluid elements can be generated by enforcing coincidence checking on the connected boundaries of the two fluid faces, while not checking the rest of the faces. However, other methods that are analogous to those in the two-dimensional case can also be used. For example, by slightly changing the shape of one face, the two faces and the open boundary part would be moved apart by a small distance while keeping the connected boundaries together. The distance between the two faces must be small enough (for example, less than ten percent of the boundary element length), but must be larger than the tolerance used in coincidence checking.

9.8 Material models

Any fluid materials that can be used in separate fluid models can also be used for fluid-structure interaction models.

Any solid materials that can be used in separate solid models can also be used for fluid-structure interaction models.

9.9 Procedure of model preparation and testing

A fluid-structure interaction model is a combined model that includes a fluid model and a solid model. They are connected through pairs of fluid-structure interface conditions. It is understood that an FSI problem cannot be solved if the individual fluid or solid model cannot be analyzed. Therefore, an essential step towards a successful FSI computation is to have the two individual models well-tested prior to coupling them together. A good reason to perform the testing in separate models is that each problem must be well-defined and no input errors should be present.

The solid model can be tested separately by applying a pressure condition on the FSI boundary to simulate the fluid stress. FSI conditions must be taken out in this test (by doing so, the solid model is separated from the fluid model). The magnitude of the pressure should be in the range of the possible fluid stresses that would be obtained in the FSI model. The element, mesh, material data, etc., have to be tested well in this model (refer to the ADINA Theory and Modeling Guide for the model preparation and testing).

To test the fluid model, apply moving wall conditions on the FSI boundary to simulate the displacements of the structure. Similarly, FSI conditions must be taken out in this test. The range of boundary movement should be as expected by the solid displacements. In addition to the guidelines in Section 13.1 on model preparation and testing, the moving mesh control needs special attention, particularly when large displacements are encountered. Refer to the guidelines on moving mesh control in Section 13.2 as well.

Once the fluid and solid models have been solved separately, the two models can be coupled by having small input changes. Remember that the control parameters of the coupled system are all specified in the fluid model. The solid model should have minimum changes. In the solid model, only the FSI condition is used to replace the pressure condition. In the fluid model, replace the moving wall condition to a corresponding FSI condition. In this stage, we focus on the “interaction” part, rather than on the conditions related to the individual models. We have to, therefore, avoid making additional changes in the coupled model, except those that are related to coupling the system. In case the coupled system cannot be solved, we know that only the coupling causes the difficulties.

Some guidelines are listed below.

- In transient analyses, make sure a good initial condition is used. This condition is more crucial in FSI computations than in individual model solutions. Since the solid variables are displacements and the fluid variables are velocities, etc., the initial conditions may not be compatible. For example, recalling that the velocity is the time derivative of the displacement, a steady-state solution of fluid velocity does not correspond to a constant displacement. When convergence difficulties arise in the first time step, it is possible that the initial conditions are not well set.
- If a pre-stretched structure is required, perform an artificial first time step, in which the structure is stretched while the fluid conditions are not yet applied (for example, zero inlet velocity, zero normal traction, etc.) as controlled in the associated time functions. In this step, the fluid mesh must be moved accordingly. The actual transient solution starts then at the second time step.
- Compute a steady-state solution as the initial condition if it is physically correct. This can be achieved using a large time step size in the first time step. Another clearer procedure is to perform two runs. The first run is to obtain a steady-state solution (may be in a few time steps). The second run is the restart execution of the first run using transient analysis conditions.
- It can be effective too to use a similar solution, as an initial condition, from a mapping file.
- The time step size must be properly determined as well. In steady-state analyses, the time step size corresponds to the increments defined in loads and boundary conditions. Therefore the time step size associated with the time functions are important. In transient analyses, the time step sizes must be small enough to capture the resolution of the solution and large enough to allow the simulation to be completed in a reasonable computer time. The general principles and suggestions are described in Section 13.1. A special indicator of proper time step sizes for FSI problems can be obtained from a frequency analysis of the individual solid model. Since the fluid equations are always nonlinear, no frequency analyses can be performed on FSI coupled models.
- There are stress and displacement convergence criteria for FSI iterations. The default option is to have both criteria satisfied. When

the displacements are very small or very large, the displacement criterion should not be used. Similarly, when the stress is very small, the stress criterion should not be used. In general, since the fluid forces are not too small in FSI analyses, the stress criterion is preferred. Note that, in iterative computing of two-way couplings, the stress and displacement tolerances should be larger in general than the tolerances used for equilibrium iterations used in the individual fluid and solid models.

- In two-way couplings, if the convergence is not achieved in the FSI iterations, you can adjust the relaxation factors for the stresses and displacements. The stress relaxation factor should be considered first since it is usually the main source of convergence difficulty. The factors should always be in the range from 0 to 1. In general, the closer to 1 the factors are, the less the number of FSI iterations should perform (if the convergence is achieved). On the other hand, the closer to 0 the factors are, the greater the possibility of reaching convergence. Large solution times are caused by factors being too small. The optimal values should be obtained from numerical experiments, preferably in a coarse mesh model.
- In 2D planar FSI using beams, we recommend using the plane strain iso-beam with unit thickness. This is because the fluid flow domain corresponds to a unit thickness slice in the units selected. If the plane stress or Hermitian beams are used, the stiffness of the iso-beam must correspond to a unit thickness (i.e., the unit thickness implicitly assumed in the 2D fluid flow).

9.10 Other types of fluid-structure interactions

In this section, we introduce more features on fluid and structural interactions. Beside the interactions on fluid-solid interface, fluid and structure can also be coupled in porous media or through thermal materials.

Consider a general fluid-structure-interaction model illustrated in Fig. 9.12. The whole computational domain Ω is divided into three regions: the pure fluid region Ω_f, a porous medium region Ω_p, and a pure solid region Ω_s. Typical structural solution variables are displacements while typical fluid solution variables are pressure, velocity and temperature. In

Ω_f, all fluid solution variables are considered. The displacement of the boundary where it abuts the solid is also a variable that is formulated in the solid model. The displacement inside the domain is arbitrary and is formulated in the ALE coordinate used in the fluid governing equations. In Ω_p, all fluid solution variables and the displacement are present, thus the fluid and solid models sharing this domain. Finally in region Ω_s, the displacement is present and the temperature variable is also included if the thermal effect on the structure is significant. The interactions thus happen in the regions where the two models are met. More precisely, the coupled regions are the interface $S\left(=\Omega_f \cap \left(\Omega_s \cup \Omega_p\right)\right)$ where traction and displacement conditions must be satisfied; the region Ω_p where stress condition must be satisfied; and finally the region Ω_s where thermal-mechanical condition must be satisfied. The three types of interactions will be called respectively, (conventional) Fluid-Structural Interaction (FSI), Porous Fluid-Structural Interaction (PFSI) and Thermal Fluid-Structural Interaction (TFSI). The coupled domain is then $\Omega_{FSI} = S \cup \Omega_p \cup \Omega_s$.

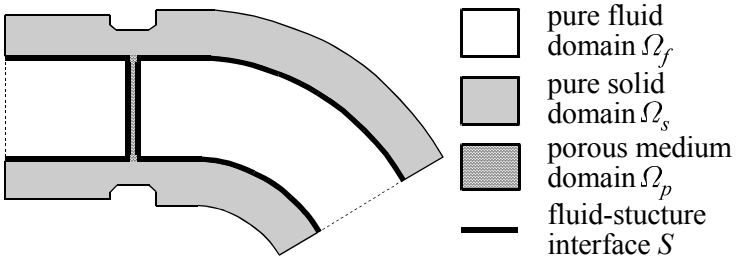


Figure 9.12 Illustration of overall fluid-structure interactions. The coupled region is $\Omega_{FSI} = S \cup \Omega_p \cup \Omega_s$

Currently, only the iterative method is applicable to PFSI and TFSI. For the computational procedure, refer to Section 9.3.

9.10.1 Fluid-solid interaction in porous media

Unlike on the interface, the fluid and structural variables are coupled in the whole porous medium. The microscope fluid stress must be added to structural model as additional internal stress $\boldsymbol{\tau}_f \left(\equiv -p\mathbf{I}\right)$. The total Cauchy stress in solid model becomes

$$\boldsymbol{\tau}_s^{total} = \boldsymbol{\tau}_s - p\mathbf{I}$$

This equation is incorporated with the finite element procedure in solid model, of course, in various elements and various structural materials.

On the other hand, the displacement compatibility between fluid mesh and solid mesh is not required inside the medium (although it is must be satisfied on the interface described in the previous section). However, as an effective and convenient method, we still use

$$\mathbf{d}_f = \mathbf{d}_s$$

in the whole porous medium. This condition also indicates that the moving mesh velocity in ALE formulation is identical with the solid velocity

$$\mathbf{w} = \dot{\mathbf{d}}_s$$

We note that the fluid mesh and solid mesh can be entirely different. Interpolation is applied to the entire porous domain to obtain, respectively, the fluid pressure on solid mesh and fluid nodal displacements.

The porosity that indicates the fluid volume percentage in the total volume is no longer a constant in coupled porous media. Assuming that the volume change of the solid part is much smaller than that in fluid part (or the so called incompressible solid skeleton), the current porosity can be obtained

$$\phi = 1 - \frac{J^0}{J}\left(1 - \phi^0\right)$$

where J is the geometric element Jacobian and the superscript 0 indicates the quantities at initial reference configuration. Accordingly, the current fluid permeability is

$$\boldsymbol{\kappa} = \frac{\phi}{\phi^0}\boldsymbol{\kappa}^0$$

9.10.2 Coupled thermal fluid-structural interaction

Thermal-mechanical interaction occurs in the solid region if structural materials are temperature-dependent (taking rubber as an example). It can also occur in porous media providing that the solid skeleton is temperature-dependent as well. The thermal condition imposed on structure model is realized through temperature-dependent materials

$$m_s = m_s\left(\theta_f\right)$$

where, m_s represents any temperature-dependent material properties in structure. The temperature is a solution variable of the energy equation and can be modeled either in the fluid or solid models.

Similar to porous media, the fluid nodal displacement is arbitrary, as long as the boundary displacement compatibility is satisfied. For convenience, we still use $\mathbf{d}_f = \mathbf{d}_s$ to update fluid mesh and, accordingly, fluid velocity in solid region is equal to the solid velocity, $\mathbf{v}_f = \dot{\mathbf{d}}_s$.

There are two methods to perform TFSI computation:

(1) The standard two-way TFSI coupling is naturally included in two-way FSI computation through standard setting. Temperature variable is included in both fluid and solid models and coupled by forcing heat-flux balance on FSI interface. This method accounts for heat generation both in the structure and the fluid. That is, heat generation due to plastic deformation or viscous effects in the structure, and heat generation due to frictional contact are now accounted for in the two-way thermal FSI analysis.

(2) The conjugate heat transfer is different from the standard TFSI by applying temperature variable only in fluid model. The fluid model must include solid element groups that are coincident with the structural domain. Temperature is defined in both fluid and solid element groups. In the case that the mesh in solid model is different from the fluid solid element-group mesh, an interpolation method is automatically applied to mapping variables that required. In order to compute the thermal effect accurately when shell (in 3D problems) or beam (in 2D problems) is used in TFSI analyses, the special boundary condition called "shell-thermal" should be applied to the fluid-shell interface. With this condition, ADINA-F can accurately predict the temperature and its gradient on shells and beams and then on nodal deformations.

9.10.3 Model one-way thermally coupled problem using mapping

In one-way thermally coupled models, it is assumed that structural deformation has negligible effect on the fluid solution. To model those problems, fluid solutions can be solved first without solving solid variables. The solutions, including temperature are saved in fluid restart file. The temperature solution is then mapped onto the solid mesh in the pre-processing stage while the solid model is created. In the solid model, temperature becomes a given condition.

Note that if the simulation times used in the fluid and solid models are different, a linear interpolated temperature between two time steps is used. For more accurate solutions, if temperature is required at multiple times in the solid model, the fluid solutions at those times should be properly saved into restart files.

This page intentionally left blank

Chapter 10 Elements

10.1 Boundary line elements

The line elements are not directly available to users. They are only used as boundary elements of two dimensional field elements. The boundary elements are automatically generated by ADINA-F. Many boundary conditions are formulated using boundary elements. Based on the elements used, two types of line elements will be encountered: a 2-node linear element and a 3-node parabolic element (see the figure below).

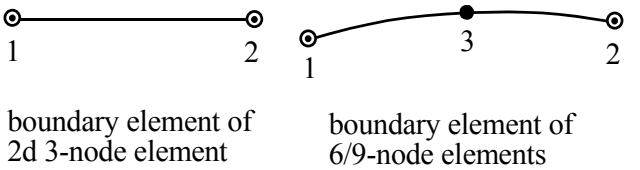


Figure 10.1 Line elements

In line elements, any variable f is interpolated as

$$f = \sum_{i=1}^{2} h_i^0 f_i \qquad \textit{for 2-node linear element}$$

$$f = \sum_{i=1}^{3} H_i^0 f_i \qquad \textit{for 3-node parabolic element}$$

where

$$\begin{aligned} &h_1^0 = 1 - r, \quad h_2^0 = r \\ &H_1^0 = h_1^0(h_1^0 - h_2^0), \quad H_2^0 = h_2^0(h_2^0 - h_1^0), \quad H_3^0 = 4h_1^0 h_2^0 \end{aligned} \tag{1.13}$$

Numerical integrations are unnecessary since all integrations can be computed accurately in analytical forms. For example,

$$\left(\nabla h_i^0\right) = \frac{\mathbf{x}_2 - \mathbf{x}_1}{\left\|\mathbf{x}_2 - \mathbf{x}_1\right\|^2}(-1,1)$$

$$\left(\nabla H_i^0\right) = \frac{\mathbf{x}_2 - \mathbf{x}_1}{\left\|\mathbf{x}_2 - \mathbf{x}_1\right\|^2}\left(h_2^0 - 3h_1^0, 3h_2^0 - h_1^0, 4h_1^0 - 4h_2^0\right)$$

$$\overline{h_1^{0n} h_2^{0m}} = \int_0^1 h_1^{0n} h_2^{0m} dr = \frac{n!m!}{(n+m+1)!}$$

and in particular,

$$\left(\overline{h}_i^0\right) = \left(\tfrac{1}{2}, \tfrac{1}{2}\right), \quad \left(\overline{h_i^0 h_j^0}\right) = \begin{bmatrix} \frac{1}{3} & \frac{1}{6} \\ sym. & \frac{1}{3} \end{bmatrix}$$

$$\left(\overline{H}_i^0\right) = \left(\tfrac{1}{6}, \tfrac{1}{6}, \tfrac{2}{3}\right), \quad \left(\overline{H_i^0 H_j^0}\right) = \begin{bmatrix} \frac{2}{15} & -\frac{1}{30} & \frac{1}{15} \\ & \frac{2}{15} & \frac{1}{15} \\ sym. & & \frac{8}{15} \end{bmatrix}$$

10.2 2D triangular element (3-node)

The 3-node triangular element can be used for two-dimensional planar and axisymmetric flows. The following figure shows the 3-node triangular element. Note that all variables are defined at the corner nodes. The center node is an "auxiliary" node where only the velocity (called bubble velocity) is defined. The center velocity is locally condensed in the assemblage so the degrees of freedom do not appear in the final equations.

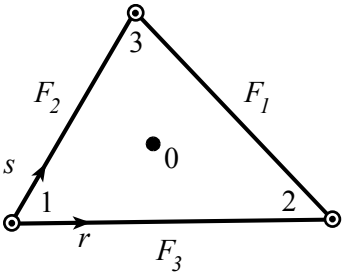


Figure 10.2 2D/axisymmetric 3-node element

The triangular element satisfies the inf-sup condition of stability and optimality. It is a linear element with first order accuracy in spatial interpolation. The elements can be generated using a free-form mesh generator for complicated geometries. The element can be used for high and low Reynolds/Peclet number flows.

In axisymmetric flows, the model corresponds to one radian of the physical domain (see Fig.2.2). Hence, when concentrated loads are applied, they must also refer to one radian.

The interpolation functions for 2D 3-node triangular element are defined as

$$h_1 = 1 - r - s, \quad h_2 = r, \quad h_3 = s, \quad h_0 = h_1 h_2 h_3 \qquad (1.14)$$

The solution variables are interpolated as

$$f = \begin{cases} \sum_{i=0}^{3} h_i f_i & if \ f = \mathbf{v} \\ \sum_{i=1}^{3} h_i f_i & else \end{cases}$$

Numerical integrations are unnecessary since all integrations can be computed accurately in analytical forms. For example,

$$\nabla h_i = -J^{-1}\mathbf{S}_i \qquad i = 1,2,3$$

$$\overline{h_1^n h_2^m h_3^k} = 2\int_0^1 \int_0^r h_1^n h_2^m h_3^k \, ds dr = \frac{2\,n!m!k!}{(n+m+k+2)!}$$

and in particular,

$$\left(\overline{h}_1,\overline{h}_2,\overline{h}_3,\overline{h}_0\right)=\left(\frac{1}{3},\frac{1}{3},\frac{1}{3},\frac{1}{60}\right),\quad \left(\overline{h_i h_j}\right)=\begin{bmatrix} \frac{1}{6} & \frac{1}{12} & \frac{1}{12} & \frac{1}{180} \\ & \frac{1}{6} & \frac{1}{12} & \frac{1}{180} \\ & & \frac{1}{6} & \frac{1}{180} \\ sym. & & & \frac{1}{2520} \end{bmatrix}$$

here, J is the Jacobian (twice the triangle area), $\mathbf{S}_i$ is the outward facial vector of the side opposite the local node i.

10.3 2D quadrilateral element (9-node)

The 9-node quadrilateral element can be used for two- dimensional planar and axisymmetric flows. The following figure shows the 9-node quadrilateral element. The pressure is defined at the four corner nodes and all other variables are defined at all 9 nodes.

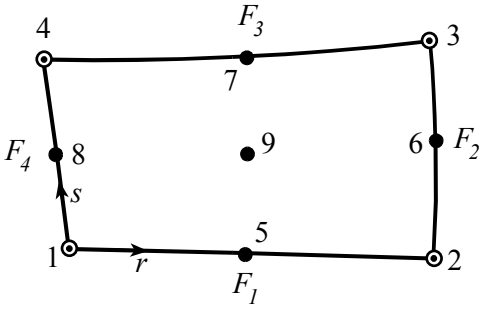


Figure 10.3 2D/axisymmetric 9-node element

The quadrilateral element satisfies the inf-sup condition of stability and optimality. The element is of second order accurate in the spatial

interpolations. The element is usually used for relatively low Reynolds/Peclet number flows.

In axisymmetric flows, the model corresponds to one radian of the physical domain (see Fig.2.2). Hence, when concentrated loads are applied, they must also refer to one radian.

The interpolation functions are H_i $(i=1,...,9)$ if all 9 nodes are included and h_i $(i=1,...,4)$ if only 4 corner nodes are included. They are defined as

$$\begin{aligned}
\left(H_1,H_2,H_5\right)&=\left(H_1^0(r),H_2^0(r),H_3^0(r)\right)H_1^0(s)\\
\left(H_4,H_3,H_7\right)&=\left(H_1^0(r),H_2^0(r),H_3^0(r)\right)H_2^0(s)\\
\left(H_8,H_6,H_9\right)&=\left(H_1^0(r),H_2^0(r),H_3^0(r)\right)H_3^0(s)\\
\left(h_1,h_2\right)&=\left(h_1^0(r),h_2^0(r)\right)h_1^0(s)\\
\left(h_4,h_3\right)&=\left(h_1^0(r),h_2^0(r)\right)h_2^0(s)
\end{aligned}$$

and the H^0's and h^0's are given in Eq.(1.13). The solution variables are interpolated as

$$f=\begin{cases}\sum_{i=1}^{4}h_i f_i & if\ f=p\\ \sum_{i=1}^{9}H_i f_i & else\end{cases}$$

No analytical integrations are available for 9-node elements. Numerical Gauss integrations are used.

10.4 2D triangular element (6-node)

The 6-node triangular element can be used for two- dimensional planar and axisymmetric flows. The following figure shows the 6-node triangular element. This element is generated as a collapsed 9-node element. The pressure is then only defined at the three corner nodes and all other variables are defined at all 6 nodes.

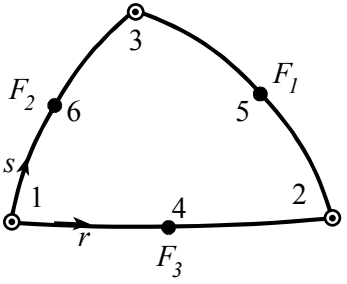


Figure 10.4 2D/axisymmetric 6-node element

The 6-node triangular element satisfies the inf-sup condition of stability and optimality. It is second order accurate in the spatial interpolation. The element is usually used for relatively low Reynolds/Peclet number flows.

In axisymmetric flows, the model corresponds to one radian of the physical domain (see Fig.2.2). Hence, when concentrated loads are applied, they must also refer to one radian.

The interpolation functions are H_i $(i = 1,...,6)$ if all 6 nodes are included

$$H_i = \begin{cases} h_i\left(2h_i - 1\right) & i = 1,2,3 \\ 4h_0 \big/ h_{k_i} & i = 4,5,6, \quad k_i = \mathrm{mod}\left(i+1,3\right)+1 \end{cases}$$

where, the h's are given in Eq.(1.14). If only 3 corner nodes are included, the linear interpolation functions defined in Eq.(1.14) are used. The solution variables are interpolated as

$$f = \begin{cases} \sum_{i=1}^{3} h_i f_i & if\ f = p \\ \sum_{i=1}^{6} H_i f_i & else \end{cases}$$

No analytical integrations are available for 6-node elements. Numerical Gauss integrations are used.

10.5 3D tetrahedral element (4-node)

The 4-node tetrahedral element can be used for three-dimensional flows. The following figure shows the 4-node tetrahedral element. All variables are defined at the corner nodes. The center node is an "auxiliary" node where only the velocity (called bubble velocity) is defined. The center velocity is locally condensed out prior to the assemblage so the degrees of freedom do not appear in the final equations.

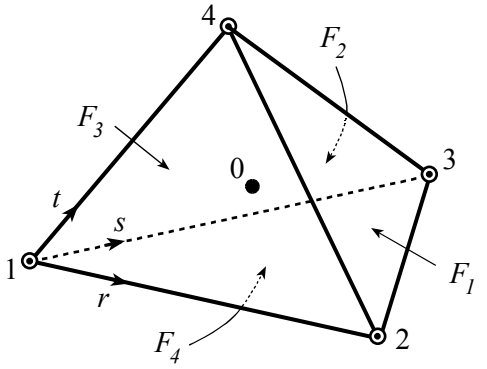


Figure 10.5 3D 4-node element

The tetrahedral element satisfies the inf-sup condition of stability and optimality. It is a linear element with first order accuracy in the spatial interpolations. The elements can be generated using a free-form mesh generator for complicated geometries. This element can be used for relatively high and low Reynolds/Peclet number flows.

The interpolation functions for 4-node tetrahedral element are

$$h_1 = 1 - r - s - t, \quad h_2 = r, \quad h_3 = s, \quad h_4 = t,$$
$$h_0 = h_1 h_2 h_3 h_4$$

The solution variables are interpolated as

$$f = \begin{cases} \sum_{i=0}^{4} h_i f_i & \text{if } f = \mathbf{v} \\ \sum_{i=1}^{4} h_i f_i & \text{else} \end{cases}$$

Numerical integrations are unnecessary since all integrations can be computed accurately in analytical forms. For example,

$$\nabla h_i = -J^{-1}\mathbf{S}_i \qquad i = 1,2,3,4$$

$$\overline{h_1^n h_2^m h_3^k h_4^l} = 6\int_0^1 \int_0^{1-r} \int_0^{1-r-s} h_1^n h_2^m h_3^k h_4^l \, dt\,ds\,dr$$

$$= \frac{6\,n!\,m!\,k!\,l!}{(n+m+k+l+3)!}$$

and in particular,

$$\left(\overline{h}_1, \overline{h}_2, \overline{h}_3, \overline{h}_4, \overline{h}_5\right) = \left(\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{840}\right), \quad \left(\overline{h_i h_j}\right) = \begin{bmatrix} \frac{1}{10} & \frac{1}{20} & \frac{1}{20} & \frac{1}{20} & \frac{1}{3360} \\ & \frac{1}{10} & \frac{1}{20} & \frac{1}{20} & \frac{1}{3360} \\ & & \frac{1}{10} & \frac{1}{20} & \frac{1}{3360} \\ sym. & & & \frac{1}{10} & \frac{1}{3360} \\ & & & & \frac{1}{415800} \end{bmatrix}$$

here, J is the Jacobian (which is six times the tetrahedral volume), $\mathbf{S}_i$ is the outward surface vector of the surface opposite the local node i.

10.6 3D brick element (27-node)

The 27-node brick element can be used for three-dimensional flows. The following figure shows the 27-node brick element. The pressure is defined at the eight corner nodes and all other variables are defined at all 27 nodes.

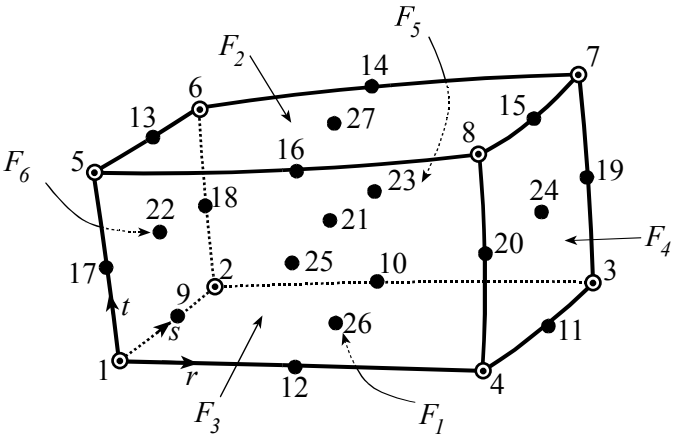


Figure 10.6 27-node 3D element

The brick element satisfies the inf-sup condition of stability and optimality. It has second order accuracy spatially while the variables are interpolated. This element can be used for relatively low Reynolds/Peclet number flows.

The interpolation functions are H_i $(i=1,...,27)$ if all 27 nodes are included and h_i $(i=1,...,8)$ if only 8 corner nodes are included. They are defined as

$$\left(H_1\ ,H_4\ ,H_{12}\right)=\mathbf{H}^0(r)H_1^0(s)H_1^0(t)$$
$$\left(H_5\ ,H_8\ ,H_{16}\right)=\mathbf{H}^0(r)H_1^0(s)H_2^0(t)$$
$$\left(H_{17},H_{20},H_{25}\right)=\mathbf{H}^0(r)H_1^0(s)H_3^0(t)$$
$$\left(H_2\ ,H_3\ ,H_{10}\right)=\mathbf{H}^0(r)H_2^0(s)H_1^0(t)$$
$$\left(H_6\ ,H_7\ ,H_{14}\right)=\mathbf{H}^0(r)H_2^0(s)H_2^0(t)$$
$$\left(H_{18},H_{19},H_{23}\right)=\mathbf{H}^0(r)H_2^0(s)H_3^0(t)$$
$$\left(H_9\ ,H_{11},H_{26}\right)=\mathbf{H}^0(r)H_3^0(s)H_1^0(t)$$
$$\left(H_{13},H_{15},H_{27}\right)=\mathbf{H}^0(r)H_3^0(s)H_2^0(t)$$
$$\left(H_{22},H_{24},H_{21}\right)=\mathbf{H}^0(r)H_3^0(s)H_3^0(t)$$

$$
\begin{aligned}
\left(h_1, h_4\right) &= \mathbf{h}^0(r) h_1^0(s) h_1^0(t) \\
\left(h_5, h_8\right) &= \mathbf{h}^0(r) h_1^0(s) h_2^0(t) \\
\left(h_2, h_3\right) &= \mathbf{h}^0(r) h_2^0(s) h_1^0(t) \\
\left(h_6, h_7\right) &= \mathbf{h}^0(r) h_2^0(s) h_2^0(t)
\end{aligned}
$$

where

$$
\begin{aligned}
\mathbf{H}^0(r) &= (H_1^0(r), H_2^0(r), H_3^0(r)) \\
\mathbf{h}^0(r) &= (h_1^0(r), h_2^0(r))
\end{aligned}
$$

and the H^0's and h^0's are given in Eq.(1.13). The solution variables are interpolated as

$$
f = \begin{cases} \sum_{i=1}^{8} h_i f_i & \text{if } f = p \\ \sum_{i=1}^{27} H_i f_i & \text{else} \end{cases}
$$

No analytical integrations are available for 27-node elements. Numerical Gauss integrations are used.

10.7 2D FCBI elements (4- and 3-node)

2D FCBI elements include 4-node and 3-node elements. They can be used for two-dimensional planar and axisymmetric flows. The following figures show the definitions of FCBI 2D elements.

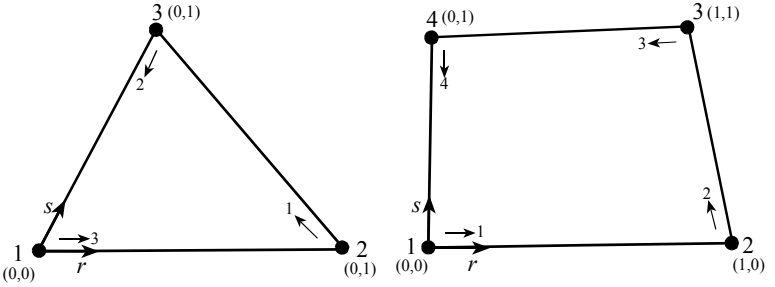


Figure 10.7 FCBI 2D elements

All solution variables are defined at corner nodes. Since step functions are used for weighting functions, FCBI elements are eventually equivalent to their counterpart: finite volume methods. Each element is thus equally divided into sub-control-volumes and integrations on both faces and surfaces are performed within elements. A flow-condition-based-interpolation function for velocity on a face, say from point $(r,s)=(\frac{1}{2},0)$ to point $(r,s)=(\frac{1}{2},\frac{1}{2})$ in a 4-node element, is defined as

$$\begin{bmatrix} h_1^v & h_4^v \\ h_2^v & h_3^v \end{bmatrix} = \left[\mathbf{h}\left(x^1\right) \quad \mathbf{h}\left(x^2\right)\right]\mathbf{h}(s)\mathbf{h}^T(s)$$

with

$$x^k = \frac{e^{q^k r}-1}{e^{q^k}-1}, \quad q^k = \frac{\rho\overline{\mathbf{v}}\bullet\Delta\mathbf{x}^k}{\mu}, \quad \mathbf{h}(y)=\begin{bmatrix}1-y\\ y\end{bmatrix}$$

where $\Delta\mathbf{x}^1=\mathbf{x}_2-\mathbf{x}_1$, $\Delta\mathbf{x}^2=\mathbf{x}_3-\mathbf{x}_4$ and $\overline{\mathbf{v}}$ is the average velocity on the face. With these functions, upwinding-effect is automatically captured in a natural way.

On the other hand, pressure, temperature and coordinates are interpolated using linear or bi-linear functions. In 4-node element, these functions are the same as defined in Section 10.3. In 3-node element, they are the same as defined in Section 10.2. In order to satisfy the Inf-sup

condition, the mass flux along face is computed from a locally condensed momentum equation.

10.8 3D FCBI elements (8-, 4-, 6- and 5-node)

These elements can be used for three- dimensional flows. The following figures show the definitions of FCBI 3D elements.

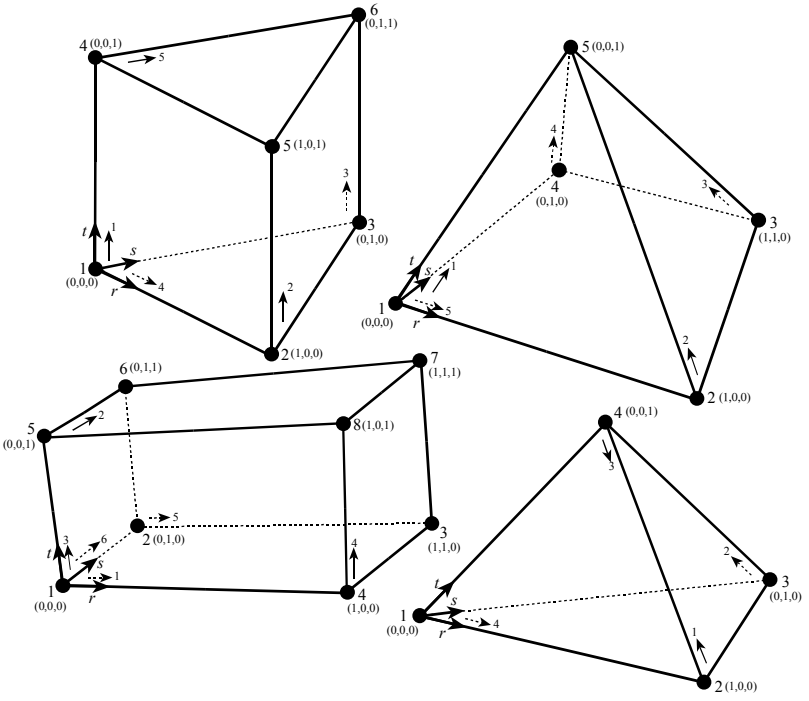


Figure 10.8 FCBI 3D elements

All variables are defined at corner nodes. The interpolations functions for velocity are similar to those used for FCBI 2D elements. The interpolation functions for pressure, temperature and coordinates are linear or bi-linear. In 8-node and 4-node elements, they are the same as defined in Section

10.6 and Section 10.5 respectively. In 5- and 6-node elements, they are defined respectively as

$$\left(h_i\right)=\begin{bmatrix}\left(1-t\right)\begin{bmatrix}\left(1-r\right)\left(1-s\right)\\ r\left(1-s\right)\\ rs\\ \left(1-r\right)s\end{bmatrix}\\ t\end{bmatrix}$$

and

$$\begin{bmatrix}h_1 & h_4\\ h_2 & h_5\\ h_3 & h_6\end{bmatrix}=\begin{bmatrix}1-r-s\\ r\\ s\end{bmatrix}\begin{bmatrix}1-t & t\end{bmatrix}$$

10.9 FCBI-C elements

These elements consist of 2D triangle and quadrilateral, and 3D tetrahedron, pyramid, prism and brick. All degrees of freedoms are defined at the center of elements. Solution variables are assumed to be piecewise constant in elements during computation, while the final solution is interpolated at corner nodes for post-processing purpose. The following figures show the definitions of all FCBI-C elements.

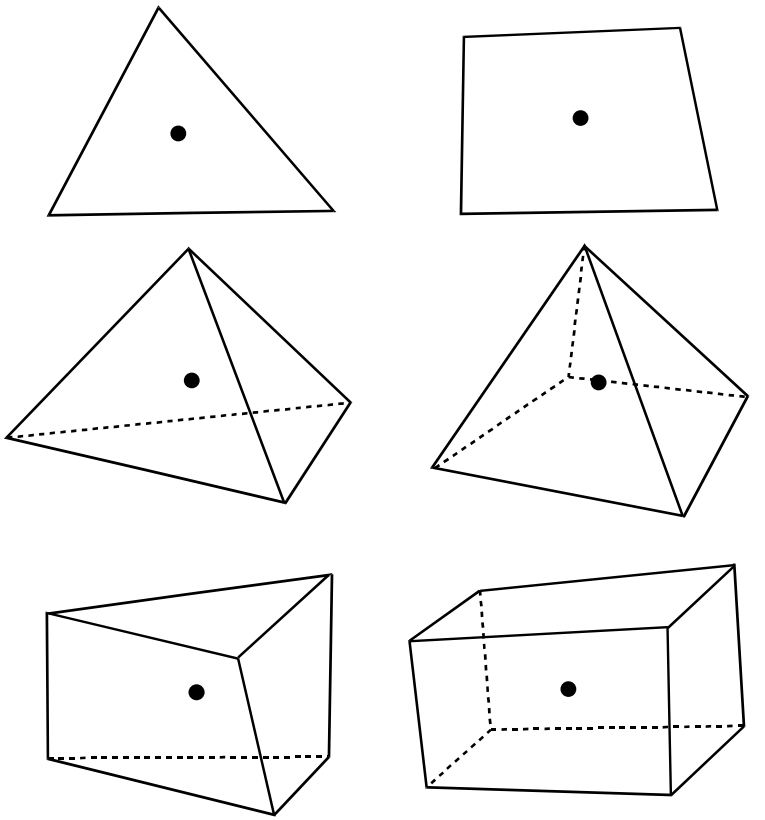


Figure 10.9 FCBI-C elements

FCBI-C elements are used for incompressible, slightly compressible and low-speed compressible flows. Only segregated method is used for FCBI-C elements.

Since the degrees of freedom are defined at the center of elements, it is natural that the following options are not applicable to FCBI-C element: constraint condition, skew system and specular radiation.

This page intentionally left blank

Chapter 11 Solutions of nonlinear equations

Most finite element or finite volume equations for fluid problems are nonlinear. The solution of a nonlinear equation must be obtained iteratively by solving some linearized algebraic equations. Iteration at this level is called outer iteration. In ADINA-F, we provide two methods in outer iteration: the Newton-Raphson and Segregated methods. The segregated method is only used for FCBI-C elements, while the Newton-Raphson method is used for other elements. For linearized equations, we provide direct sparse solver and some iterative solvers. The iteration in iterative solvers is called inner iteration. These nonlinear equation methods and linear solvers are described in this chapter.

11.1 Newton-Raphson method

We use the Newton-Raphson method in ADINA-F for all types of elements except the FCBI-C elements. This method is usually used for the system of all solution variables, although it may be modified using a better solution strategy for certain problems. The user needs take no action in respect of the modification.

A typical procedure is summarized here. Consider a nonlinear equation $f(x)=0$, where x is the solution vector that includes all active solution variables $x=(x_v)=(u,v,w,p,\theta,...)$. We take the solution at time t as the initial guess of the solution x^0 at $t+\Delta t$. The following equilibrium iteration procedure is performed for obtaining the solution:

$$\text{for } k=1,2,\ldots \begin{cases} \text{solve for } \Delta x^k : \left(\partial f^{k-1}/\partial x\right)\Delta x^k = -f\left(x^{k-1}\right) \\ x^k = x^{k-1} + \Delta x^k \\ r = \max_v \left\{ \left\| \Delta x_v^k \right\| \Big/ \max\left\{ \left\| x_v^k \right\|, 10^{-8} \right\} \right\} \\ \text{if } \left(r \le \varepsilon\right) \text{ stop} \end{cases}$$

where ε is the tolerance (its default value is 10^{-3}).

Depending on the type of problem, the overall outer iteration may be altered, say, combined with a few Newton-Raphson iterations for the solution system. One example is the iterative coupling of fluid-structure interation, where the fluid and solid variables are solved separately, treating the others are fixed during the iteration.

The residuals obtained after iteration 1 are also checked. If they are smaller than a steady-state tolerance ε_s, the program stops since the steady-state solution has been obtained. Otherwise, the program continues for the solution at the next time step.

11.2 Segregated method

The segregated method is an iterative algorithm for solving the nonlinear fluid system. It is used here only for FCBI-C elements. It is natural that the segregated method cannot be used for direct FSI coupling. This method also cannot be used for specular radiation problems.

With the exception of continuity, all linear equations are derived directly by integrating each particular variable regarding all other variables as fixed. For continuity equation, the discretized momentum equations are used to derive a pressure-correction dependency on corrections to velocity. The linear solver is then applied to each variable in a certain order in the outer iteration. In each time step, the segregated method can be typically described in the following diagram.

$$
\begin{Bmatrix}
for\ outer\text{-}iteration\ \left(k=1,K_{\max}\right) & & \\
\left\{\begin{array}{l}
for\left(k_{fluid}=1,K_{fluid}\right) \\
\left\{\begin{array}{l}
for\left(k_{vp\theta}=1,K_{vp\theta}\right)\left\{\begin{array}{l}
for\left(k_{vp}=1,K_{vp}\right) \\
\left\{\begin{array}{l} solve\ for\ u^k,v^k,w^k,p^k\ \ one\ by\ one \\ if\left(converged\right)break \end{array}\right.
\end{array}\right. \\
solve\ for\ \theta^k \\
if\left(converged\right)break
\end{array}\right. \\
for\left(k_{K\varepsilon}=1,K_{K\varepsilon}\right)\left\{\begin{array}{l} solve\ for\ K^k\ \ and\ then\ for\ \varepsilon^k/\omega^k \\ if\left(converged\right)break \end{array}\right. \\
if\left(converged\right)break
\end{array}\right. \\
solve\ for\ other\ variables\ one\ by\ one \\
solve\ for\ d^k_{solid} \\
for\left(k_d=1,K_d\right)\left\{\begin{array}{l} solve\ for\ d^k_{fluid} \\ if\left(converged\right)break \end{array}\right. \\
if\left(converged\right)break
\end{Bmatrix}
$$

Each "solve for …" step involves the solution of linearized equations. This solution can be obtained by use of either the direct sparse solver or the iterative AMG1, AMG2 or RPBCG solvers.

Since the solution variables are not coupled directly in one matrix system, it is essential that the solution be converged, or else a non-physical solution may be obtained. If an iterative solver is used for the linear equations, it is also important to properly control the convergence in the inner iterations. In the next sub-sections, these issues will be discussed.

11.2.1 Basic parameters

The coupling of velocity and pressure is one of the important steps towards a converged solution. *Coupling schemes* we used are either SIMPLE (default) or SIMPLEC. In SIMPLE method, usually two under-relaxation factors, λ_v for velocity and λ_p for pressure, must be applied to ensure the overall convergence. In SIMPLEC method, the velocity relaxation factor is usually unnecessary and can be set to one.

The FCBI-C element is "naturally" first order in space since all variables are defined at the element centers. However, it can become second order in space. *Scheme order* provides the choice of first (default) or second order accuracy in integration of the flux on element face. The second order scheme is normally less stable and requires more iterations, but the solution is more accurate than the solution of the first-order with fixed number of elements. When element sizes tend to zero, the difference of the two solutions diminishes too.

The *nodal variables are interpolated* whenever they are needed in computation. The first approach is the neighboring cell-size-weighted average. The first order nodal variable can be improved to become second order (which is our default option). However, in problems with high difficulty and/or very poor mesh quality, where the second order method hardly converges, first order method may be used.

The default scheme for *pressure interpolation* at the faces uses a natural control volume approach. This procedure works well for most smooth-varying pressure problems. For problems with large body forces or second-order time integration scheme, the linear scheme should be used. In this scheme, pressure is computed using a linear interpolation of neighboring element values.

The *Pressure-Implicit with Splitting of Operators (PISO)* pressure-velocity coupling scheme is based on a higher degree of approximate in outer iterations. In general, with the choice of PISO, the solution convergence can be achieved with less iteration, but slightly more expensive in each iteration. PISO is highly recommended for turbulence models.

By default, the solver for the displacement of the moving mesh is the same as that for fluid variables. That is, the unknown displacement is defined at element center and the matrix of linear equations is non-symmetrical. The nodal displacement is then obtained using interpolation of element-center displacements. This is true even when the fluid solver is sparse solver. Until the convergence in the outer iteration, the displacement solution is usually not converged yet (with respect to current conditions) in each sub-iteration. If the intermediate mesh causes difficulty in overall convergence, one can specify more sub-iterations for displacement (K_d in the diagram). The advantage of this selection is that the program uses no additional storage in solving for the displacement.

However, if the memory is enough, the sparse solver can be directly applied to solving the nodal displacement. The *mesh solver* allows the user to select this option. Although extra storage is used, the nodal displacement is directly solved from a symmetrical matrix. In addition, if the original mesh is used as the background mesh in solving for the displacement, the matrix will be factorized only once, thus the solver uses minimum CPU time. For most moving-mesh problems, it is recommended to use the sparse solver as the mesh solver and the original mesh as the background mesh. It is more efficient, more accurate and more stable.

For some problems, it is more efficient to divide the computational procedure into a few stages. For example, in the first stage, one computes the fluid variables and, in the second stage, only computes the temperature (if the fluid is independent of temperature). The user can achieve this by defining *active-variables* in an execution. By default, active variables are all the solution variables required in the analysis. An inactive variable is involved in all computations, with the exception that it will not be solved and updated.

In certain transient problems, good initial conditions are difficult to specify. In this case, user may choose a restart run, using the initial condition that is obtained in another steady-state analysis. In the segregated method, we allow the two executions be performed in one analysis. The *first step steady-state solution* option forces the program to obtain a steady-state solution in the first time step and transient solutions in the remaining time steps. To turn on this option, the user needs to input the maximum

number of iterations specifically for the first time step. Of course, this analysis can also be performed using restart run.

Limit allows the user to specify certain extreme values. These values are minimum values of turbulence variables and maximum factor for viscosity during iteration, maximum CPU time allowed in the execution, etc.

11.2.2 Outer iteration control

11.2.2.1 Stopping criteria

The convergence criteria used in outer iteration may differ depending on the simulation being carried out. The user may want to completely or temporarily terminate an execution of the program for different reasons: solution is converged; CPU time limit is reached; being able to check the quality of an intermittent solution and adjust relaxation factors; etc. The program can be stopped in a number of ways.

Tolerances on residuals are used to identify the convergence of the solution. In segregated method, we always use a modified L_1-*normal* for an array x

$$\|x\| = \begin{cases} \|x\|_e \equiv \frac{1}{N_e}\sum_{i=1}^{N_e}|x_i| & \textit{for element quantities} \\ \|x\|_f \equiv \frac{1}{N_f}\sum_{i=1}^{N_f}|x_i| & \textit{for element-face quantities} \end{cases}$$

where N_e and N_f are number of elements and number of element-faces respectively.

Let a discretized conservation equation for a general variable v at an element be written as

$$e = a_0 v_0 + \sum_{nb} a_{nb} v_{nb} - b_0 = 0$$

A relative equation residual (ER) is computed as

$$r_e = \|e\| / e_{\text{mod}}$$

where for the continuity equation

$$e_{\text{mod}} = \max\left\{e_{\min}, \xi^e_{\min}\|e_f\|, \|e_m\|\right\}$$

and for other equations

$$e_{\text{mod}} = \max\left\{e_{\min}, \xi^e_{\min}\|e_f\|, \min\left\{\xi^e_{\max}\|e_f\|, \|e_m\|\right\}, \|e_d\|\right\}$$

In which, the parameters $e_{\min}, \xi^e_{\min}, \xi^e_{\max}$ are pre-specified, $\|e_m\|$ is the maximum residual after I_0 *initial iterations* (by default it is 5 and can be modified), and

$$\|e_f\| = \max\left\{\left\|\int_e (\textit{source-term})\, dV\right\|_e, \left\|\oint_f (\textit{flux-vector}) \bullet d\mathbf{S}\right\|_f\right\}$$

$$\|e_d\| = \left\|\sum\nolimits_{nb} |a_0 v_0|\right\|$$

A relative variable residual (VR) is computed as

$$r_v = \|\Delta v\| / v_{\text{mod}}$$

$$v_{\text{mod}} = \max\left\{v_{\min}, \xi^v_{\min}\|v\|, \min\left\{\xi^v_{\max}\|v\|, \|\Delta v_m\|\right\}\right\}$$

Similarly, $v_{\min}, \xi^v_{\min}, \xi^v_{\max}$ are pre-specified constants, $\|\Delta v\|$ is the absolute variable residual, and $\|\Delta v_m\|$ is the maximum variable residual after the initial iterations (I_0).

Let $f = v, e$ represent the variable and equation. Obviously, the pre-defined constants are problem dependent. Being different from the dimensionless constants $\left(\xi^f_{\min}, \xi^f_{\max}\right)$, $f_{\min}$ also has a specific unit. Assuming that all solution variable quantities are of order one, we initially set, for all variables and equations

$$\left(f_{\min}, \xi^f_{\min}, \xi^f_{\max}\right) = \left(10^{-5}, 1, 10\right)$$

Note that, if variable quantities are far different from order one, these constants must be changed. For example, if $p \Big/ \left(\frac{1}{2}\rho v^2\right) \gg 1$ as usually happen in low-speed compressible flows, the momentum equation residuals could become too small because its equation scale (including the magnitude of pressure) is too large. This will cause the program to skip from solving for the velocity and then converged to a wrong solution. This symptom can be observed if velocity residual is always zero. In order to solve the problem, one can simply reduce $\left(e_{\min}, \xi^e_{\min}, \xi^e_{\max}\right)$ for momentum equations such that a non-zero velocity residual is observed.

A single variable/equation f_i is said to have converged if

$$\left(r_{f_i} < \varepsilon_f\right) \& \left(k \geq K_{\min}\right)$$

where ε_f and $K_{\min}$ are, respectively, the tolerance and minimum number of iterations requested. The convergence of a variable or equation is defined as convergence of some variables or equations of the user's preference. *Variable type and equation type on tolerance* are parameters used to identify the possible combinations of variables/equations. VR/ER type can be ALL, NO, or any one the solution variable. For example, if ALL is the option for VR, the variable is said converged if all solution variables are converged. Similarly, if NO is the option for ER, all solution residuals will be ignored in convergence checking. Finally, the *overall convergence is defined as either VR or ER is converged*. The default set is ALL for VR, pressure for ER (mass conservation) and the tolerances are $\left(\varepsilon_v, \varepsilon_e\right) = \left(10^{-3}, 10^{-4}\right)$. Obviously, the ER type may differ depending on

the simulation being carried out. For instance, it may be set to either NO or temperature (energy conservation) in a pure conduction problem. Notice that, as convention, velocity, pressure and energy equations are also called force (or momentum), mass and energy conservations respectively.

Maximum number of iterations is used to stop the execution after that number of iterations for the current time step. The number of iterations required may be quite different, depending on the accuracy desired in the solution and the complexity of the problem, from a couple of iterations for simple and small transient problems, to even several thousands for some large steady-state problems. When program stops due to the specified maximum number of iterations, one must check if the solution has truly converged. If it is obviously diverging, the model must be examined, possibly modified, and executed again. If it is converging, the problem can be continued in a restart-run. Differing from a normal restart-run, one should specify the initial time the same as the last time at which the solution did not converge. If you intend to do so, simply ignore the warning message regarding the "unmatched initial time". In the model of restart runs, the relaxation factors can be slightly adjusted too, according to their history, to accelerate the convergence.

Minimum number of iterations is used to force the program to iterate at least to that number. If it is specified the same as the maximum, the program will stop after that number of iterations regardless of convergence or divergence.

CPU limit provides another option to stop the program in a certain time period. Again, the solution must be checked and, if necessary, executed in a restart run as described before.

11.2.2.2 Relaxation factors

Since the fluid variables are not fully coupled into a system (as in the Newton-Raphson method) on each iteration, under-relaxation factors are usually required. The factors may differ depending on the solution variable being solved, the difficulty of the problem and the mesh quality of the computational domain.

Considering a typical linearized equation

$$a_0 \Delta x_0 + \sum_{nb} a_{nb} \Delta x_{nb} = e_0$$

there are two types of relaxation factors that are used in the program. The variable relaxation factor (VRF) is applied to diagonal of the matrix $a_0 \rightarrow a_0 / \xi_v$. The equation relaxation factor (ERF) is applied to the right-hand side of the equation (residual) $e_0 \rightarrow \xi_e e_0$.

VRF is used to overcome the numerical difficulties raised from non-linearity and to improve the conditioning of the linear equation. It is also served equivalently as a form of pseudo-local-time step (see false time step option), i.e., a small VRF corresponds to a small time step and large VRF corresponds to a large time step. Hence a small VRF will not only lead to a faster solution of the linear equation in iterative solver, but also help on overall convergence. It is also understood that too small a VRF (like a small time step) will lead to slower global convergence.

ERF is used to overcome the overall convergence difficulty raised from a large source term in a linear equation. This source may be physically real (e.g., in turbulence equations) or a numerical error (a bad solution approximation at the start of each iteration). A small ERF adds the source incrementally in the outer iteration and, therefore, improves the global convergence.

Frequently, ERF is used for pressure while VRF is used for other variables. We treat these factors as "primary" while others as "secondary". Notice that the primary pressure-relaxation factor is applied to the equation to enforce the mass conservation from intermittent solutions.

Transient analyses differ from steady-state analyses by the presence of the dynamic term that may stabilize solution equations (except the continuity equation in incompressible flows). For small time step length (used in short-wave problems, for instance), VRF is usually unnecessary, meaning that it can be set to one. However, if the time step is not small (used in long-wave problems, for instance), the solution could diverge. In this case, the *primary factors for transient analyses* must be adjusted accordingly, as those for steady-state analyses. Since we allow mixed transient/steady analyses in a single model, both relaxation factors for steady-state and transient analyses are required.

In summarizing, we have introduced the primary factors for steady-state analyses (PRI-SS), the primary factors for transient analyses (PRI-TR) and secondary factors (SEC). Their default values are listed in the table below.

Figure 11-1 Default set of outer relaxation factors in segregated method

Variable	PRI-SS	PRI-TR	SEC
Velocity	0.75	0.75	1
pressure	0.3	0.3	1
displacement	1	1	1
turbulence-K	0.97	0.97	1
turbulence-ε/ω	0.97	0.97	1
other variables	0.99	0.99	1

11.2.2.3 Convergence information

At the end of each outer iteration, information is printed into the *.out file. A typical output in outer iteration is in the format

ITER, ER-P, max(VRs), ER-V, VR-V, VR-P, (VAR, CELL, MAX-VR, CELL, MAX-ER)

where "cell" is the global element number (counted from the first element of the element group 1 to the last element of the last element group) and

ITER = outer iteration number
ER-P/V/etc. = relative equation residuals for pressure, velocity, etc.
VR-P/V/etc. = relative variable residuals for pressure, velocity, etc.
Max(VRs) = maximum value of all relative variable residuals
VAR = variable name for which the maximum equation residual occurs
MAX-VR, CELL = maximum absolute variable residual for VAR
MAX-ER, CELL = maximum absolute equation residual for VAR

The user can choose the *variable for which the maximum residuals are printed*. The default is the variable that has the maximum absolute residual.

11.2.3 Inner iteration control

A direct or an iterative solver can be used for solving the linearized equations. Inner iteration refers to only iterative solvers. The available iterative solvers associated with the segregated method are AMG1, AMG2 and RPBCG.

One simple strategy would be forcing a full convergence in iterative solvers for every solution variable. However, since the goal of iteration is to reach overall convergence of the nonlinear equation, an exact solution is not required because this is just one step in the nonlinear outer iteration. Managing the inner iteration to a minimum level with yet a reasonably good intermittent solution may save much solution time.

11.2.3.1 Inner convergence criteria

A proper measurement of the quality of a solution leaving the iterative solver is the first step to efficient inner iteration. We consider a solution good enough if the residual in the inner iteration is smaller than a "percentage" of the critical residual in outer iteration convergence. The "percentages" used here are called variable and equation reduction numbers (σ_v and σ_e) respectively. Beside the reduction numbers, maximum number of iterations $I_{\max}$ and minimum number of iterations $I_{\min}$ can also be specified.

Consider an algebraic equation $Ax = b$ and recall that the unknown vector x and its right-hand side b are the variable and equation residuals respectively in outer iteration. Let i be the inner iteration number, we define three types of residuals in inner iteration

$$\begin{aligned} &\textit{variable residual}\,(VR): && r_v^i = \left\| x^i - x^{i-1} \right\| \Big/ v_{\text{mod}} \\ &\textit{equation residual}\,(ER): && r_e^i = \left\| Ax^i - b \right\| \Big/ e_{\text{mod}} \\ &\textit{reducted residual}\,(DR): && r_d^i = \left\| Ax^i - b \right\| \Big/ \left\| b \right\| \end{aligned}$$

where v_{mod} and e_{mod} are the variable and equation scales, respectively, determined in outer iteration. We define any one of the followings as the *convergence criterion* in inner iteration:

$$(1) \qquad r_e^0 \le \sigma_e \varepsilon_e$$

$$(2) \qquad \left(r_e^i \le 0.001\sigma_e \varepsilon_e\right) or \left(r_e^i \le \sigma_e \varepsilon_e\right) \& \left(i \ge I_{\min}\right)$$

$$(3) \qquad \left(r_d^i \le 0.001\sigma_e \right) or \left(r_d^i \le \sigma_e \right) \& \left(i \ge I_{\min}\right)$$

$$(4) \qquad \left(r_v^i \le 0.001\sigma_v \varepsilon_v\right) or \left(r_v^i \le \sigma_v \varepsilon_v\right) \& \left(i \ge I_{\min}\right)$$

where ε_v and ε_e are the tolerances defined in outer iteration. The parameters that control the solution process are different for each variable/equation. The default set is

Figure 11-2 Default set of inner relaxation factors in segregated method

Variable	σ_e	σ_v	$I_{\min}$	$I_{\max}$
pressure	0.01	0.01	2	300
velocity	0.01	0.01	2	50
others variables	0.1	0.1	2	50

We note that, if the inner convergence criterion (1) is satisfied, no linear equation solvers will be invoked (even a direct solver is selected), leaving the solution unchanged at the end of that outer iteration. It is important to enforce a tighter convergence of the continuity equation than other equations. A false inner convergence of the continuity equation will usually lead to a diverged solution in the outer iteration.

11.2.3.2 Inner convergence information

Whenever an iterative solver is invoked, inner iteration information is printed into *.log file in the format shown below, at the end of every selected number of "*outer iteration interval for print*" (it is 1 by default),

V: ITE, I, ED, ER, VR, EI, E0, EM, VI, VM, V0 = ...

where

V = abbreviation of variable name
ITE = outer iteration number
I = inner iterations performed
ED, ER, ER = r_d^i, r_e^i, r_v^i
EI, E0, EM = $\left\|Ax^i - b\right\|, \left\|b\right\|, e_{\text{mod}}$
VI, V0, VM = $\left\|x^i - x^{i-1}\right\|, \left\|x^i\right\|, v_{\text{mod}}$

At the end of model execution, the total number of inner iterations performed is printed.

11.2.4 Comments and tips on convergence

The segregated method requires far less memory than the Newton-Raphson method. The memory usage is basically linear in relationship with the number of elements if an iterative solver is used. The convergence could be very fast too (say that CPU time is linearly related to the number of elements), if the outer and inner solution processes are properly controlled. These advantages make this method very useful in solving very large problems. However, a drawback of this method is the relatively high difficulty level of having the solution fully converged. Remember that not-fully converged solutions may be physically unacceptable. For this reason, we have given users more controls of this method than the Newton-Raphson method.

Model preparation and testing is definitely the first important step towards a good solution. Experience in model analyses has shown that the most likely cause of a divergence process or of convergence to an obviously incorrect solution is that the model has not been set up as intended. If the model is believed to have a physically stable and unique solution, the error is likely in the input. For the purpose of testing the model, while keeping the time consuming part in the model (such as the geometry, element groups, etc.), one can always use the smallest and simplest model possible. For example, by reducing the number of elements, using larger viscosity and a smaller buoyant force, etc. Using

this model, the problem can be easily and quickly identified. If the input is correct, try to identify the cause of convergence difficulty.

Reducing variable relaxation factors can be an efficient way to overcome the convergence difficulty if it is not a complex problem. Usually the primary pressure relaxation factor need not be adjusted for majority of problems. Note too that small relaxation factors may lead to false convergence.

Unless there is a special purpose, *avoid using variable under-relaxation factor for pressure*, since the intermittent solution may violate the *mass conservation*–an important quantity used in other convection-diffusion equations. Notice that the variable relaxation factor for pressure is the secondary factor.

To obtain a steady-state solution of very large size, try to *break it into a few consecutive runs*. In each run, adjust control parameters if necessary based on the printed convergence history.

If the cause of divergence is due to sudden large loads (normal traction, prescribed velocity, etc.), *add the loads incrementally* in a few time steps, say in 3 time steps each of them composing 1, 11 and 89 percents of loads respectively. Usually the most difficult times are the transitions of flow patterns, for example, in the first time step, when flow from stationary changes to a complex flow pattern.

If a *steady-state analysis* is neither converging nor diverging and the situation persists even the relaxation factors are reduced, it is possible that a steady-state solution does not exist. More precisely, it may not exist for the "input conditions". A typical example is the "laminar" flow of high Reynolds number in a backward-facing step, in which the circulation reaches the outlet where a constant pressure is specified. The pressure condition that is specified improperly inside the circulation may cause the numerical oscillation. In this case, one has to use a turbulence model or a smaller Reynolds number in the laminar model if physically acceptable. Another efficient solution is to use *time-marching* technique. That is, one can perform a real transient analysis with many time steps (may be in a few runs as well), until the solution is visually unchanged. Since the final

solution is of interest, a loose tolerance can be used in outer iteration, using *equal minimum and maximum numbers* of iterations or larger tolerance.

In *transient analyses*, an immediate divergence usually indicates a large time step. Use CFL~1 as a criterion to check if the time step is small. For *small time step* as in many short-wave problems, the default set of primary relaxation factors for transient analyses is adequate. A not-fully converged solution may also be acceptable depending the accuracy requirement. On the other hand, if a *large time step* is used, as in many long-wave problems, the set of relaxation factors must be adjusted to the set for steady-state analyses. The convergence must be enforced as well in order to obtain accurate transient solutions. One may also use the *composite scheme* of 2nd order in time integration to improve the solution quality.

In a *turbulence model*, experience has revealed that *local velocity-pressure iteration* $\left(K_{vp} > 1\right)$ may improve the overall convergence. We note that $K_{vp} = 2$ corresponds to the Pressure-Implicit with Splitting Operators (PISO) method if both predictor and corrector are implicitly treated. Furthermore, *local turbulence-variable iteration* $\left(K_{K\varepsilon} > 1\right)$ may also improve the overall convergence.

In a *fluid-structure-interaction model*, a few iterations of fluid variables is necessary $\left(K_{fluid} = 2 \sim 10\right)$. For very difficult problems, a large K_{fluid} may be specified. In this case, the iteration in fluid model is controlled by the specified tolerance.

In *buoyant-driven flows*, if difficulty in convergence occurs, try to use *secondary under-relaxation factor* for velocity. This is equivalent to adding buoyant force gradually in iterations. One can also introduce a *local velocity-pressure-temperature* $\left(K_{vp\theta} > 1\right)$ iteration, say 2 to 3 times, to tighten the coupled relationship between the velocity field and the temperature. In principle, if the order of the right-hand side of an equation is much higher than the order of the magnitude of the matrix, the corresponding secondary factor may be reduced.

A *proper unit system* can be efficient in minimizing the round-off errors and reducing the chance of instability caused by round-off errors. One can either input data in a proper unit system, or choose proper nondimensional scales, such that *all solution variables are of order one*. If the linear solver is always skipped from solving a variable (*exact* zero residual is printed in the outer iteration), the residual of that equation is too small. This may be a sign that an improper system of units is used. Of course, one can also adjust the constants defined in the scale for that equation.

A mesh is said to be orthogonal if the line that connects the centers of the two neighboring elements is perpendicular to the face they share. If a 2D 4-node element or 3D 8-node is applied to a simple-geometry problem, the mesh is usually orthogonal or nearly orthogonal. For an *orthogonal mesh*, large variable relaxation factors may be used (close to 0.99~1 if possible without causing divergence) and, accordingly, fewer iterations may be performed. On the contrary, for a *free-form or distorted mesh*, smaller variable relaxation factors are usually necessary, and more iterations will be performed.

For an equation with a *dominant diffusive term*, the convergence could be very slow, unless a larger variable relaxation factor is used (set the factor close to 0.99~1 if possible without causing divergence).

For *pure conduction* problems (without the presence of a fluid field), the *mass-conservation* must not be checked, or, have the *energy equation residual* option checked.

For moving mesh problems, try to use the *original mesh* as the background mesh if possible (that means the choice of the original mesh will not cause difficulty in overall convergence). If there is enough memory, always use the sparse solver as the *mesh-solver*. The use of the two in the same model can achieve the best in speed, accuracy and convergence regarding moving mesh.

The solver *AMG1* should be always the first choice of iterative solvers. Breakdown in iterative solvers usually indicates a singular system, meaning some errors exist in the model that should be always checked first. AMG2 can be used, in very rare situations, if AMG1 has failed in inner-iterations. It uses slightly more memory.

A well-converged inner iteration is important in reliably obtaining solutions of difficult problems. This is particularly true for the pressure equation. Carefully check the information printed out in the *.log file. To enforce a full convergence, reduce the reduction numbers and increase the maximum number of iterations. For difficult problems, a much larger number of inner iterations for pressure may be required (say 400~800). Experience has showed that not-fully-converged pressure solutions in inner iteration usually result in divergence in outer iteration, while not-fully-converged velocity fields result in neither divergence nor convergence in outer iteration.

Inner iteration tends to be most difficult at initial iterations (meaning more iterations are performed) and becomes easier when solution is closer to convergence. *Too-difficult inner convergence* indicates a poor conditioning of the matrix, suggesting the application of the loads over a few time steps or reducing the variable/equation relaxation factors. On the contrary, a *too-easy inner convergence* indicates the tolerance is too loose. A tighter reduction number or a larger minimum number of iteration may be necessary.

11.2.5 Limitations

The segregated method is only used for FCBI-C elements. It cannot be used for direct FSI coupling, specular radiation, constraint condition, skew system and ATS option.

11.3 Solvers of algebraic equations

Solving a system of linear equations $Ax = b$ is essential to the computations in ADINA-F. Most of the storage assigned to the program is occupied by the matrix system and most of the CPU times used by the program is devoted to the solution of the algebraic equations. Various solvers are available for solving the linear equations that are resulted from the Newton-Raphson or Segregated methods. They are briefly introduced here.

11.3.1 Gauss elimination method

This method is probably the most famous direct solution technique. In ADINA-F, the active column solution is implemented (COLSOL). This method should not be used, in general, because the sparse solver described in the next section is much more effective. It is introduced here briefly to be able to better describe other solvers later.

In this method, the matrix structure is initially optimized to have short active columns (a low skyline) according to the finite element connectivity, the constraint relations and to have sufficient fill-ins to not encounter zero pivots. The area within the skyline stores all matrix elements before and after the equation solution.

First, the matrix is decomposed as in the Gauss elimination (without pivoting),

$$A = LDU$$

The diagonal elements of the matrices L and U are all one and the other nonzero elements are located in the lower and upper positions of column i, respectively. The matrix D is diagonal. These structures allow us to store all the matrices in the original matrix A.

Once the decomposition is complete, the solution of the equation can be obtained by:

$$X = A^{-1}b = U^{-1}D^{-1}L^{-1}b$$

Note that the matrix has never been inverted.

Calculating the complete *LDU* decomposition without taking advantage of possible "no-fill-ins" (see Section 11.2) is in general a very expensive procedure according to current standards of matrix computations.

The matrix requires storage of $N_d m_b$ words, where m_b is the averaged bandwidth.

The algorithm requires arithmetical operations of the order of $N_d m_b^2$.

Furthermore, the direct decomposition of the matrix may have a large round-off error even if the matrix is well-conditioned. For example, for the simple matrix

$$\begin{bmatrix} \varepsilon & 1 \\ 1 & \varepsilon \end{bmatrix}$$

the process fails to give the solution when $\varepsilon = 0$ if no pivoting is used (notice that the conditioning of the sample matrix is good). It is understandable then that the *LDU* decomposition of this sample matrix will have a large round-off error if ε is small.

The methods of column and/or row interchanges (pivoting) are very effective in overcoming these problems but add much computational effort. For this reason, in ADINA-F, the variables are ordered to not encounter a singular system. The solution then does not allow for row/column interchanges and hence, assumes that these are not needed.

When there is not enough memory to hold the matrix in core, this solver automatically goes out-of-core. Then the computational speed is slowed down by a factor of order n_b^2, where n_b is the number of blocks of the matrix.

11.3.2 Sparse solver

The sparse solver is a direct solver based on the Gauss elimination method. However, it differs from the traditional direct solvers in preserving the sparsity of the matrix, thus reducing dramatically the storage and computer time required by traditional direct solvers.

The assembled matrix is initially stored in compact form, according to the finite element connectivity and the constraint relations. It contains only logically nonzero elements in the assemblage level.

The sparse solver then reorders the finite element equations by keeping the sparsity of the original matrix at its best. The sparse solver also performs a symbolic factorization to locate those elements that will become nonzero during the factorization. These elements are called fill-ins. The number of fill-ins determines the additional storage that is required in the solver. The number of fill-ins together with the structure of the reordered matrix determines the effectiveness of the computations performed in factorizing the matrix.

Here is a simple example that explains how the sparse solver works. Consider a heat transfer problem with 5 degrees of freedom defined at the corresponding nodes (see the following figures).

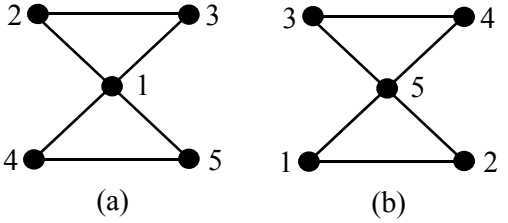


Figure 11-3 Ordering in sparse solver

Fig.11.1(a) shows a possible numbering of equations. The corresponding symbolic sparse matrix structure is

$$A = \begin{bmatrix} * & * & * & * & * \\ * & * & * & + & + \\ * & * & * & + & + \\ * & + & + & * & * \\ * & + & + & * & * \end{bmatrix}$$

where "*" indicates the original nonzero elements and "+" indicates the additional fill-ins. The matrix in this example eventually becomes full. This ordering is therefore not effective.

If the order of the equations as shown in Fig.11.1(b) is used, we have the following symbolic sparse matrix structure

$$A = \begin{bmatrix} * & * & & & * \\ * & * & & & * \\ & & * & * & * \\ & & * & * & * \\ * & * & * & * & * \end{bmatrix}$$

where the empty elements are neither stored nor computed. There are no fill-ins in this example; therefore, we have achieved the best order. This is exactly what the sparse solver aims to achieve in reordering: to minimize the number of fill-ins as shown in this example.

The minimum degree algorithm and the multi-graphics processes are the most popular schemes of symbolic factorizations used in sparse solvers. We use a hybrid of these two methods.

In addition to preserving the sparsity of the matrix system, we also apply the "super equation" concept in regrouping equations into groups, such that equations within each group have the same symbolic structure. A group is further cut into panels that are formed by blocks, whose size is determined based on the cache size of the machines used. The Gauss elimination procedure is then performed in blocks instead of in columns. This amalgamation reduces the data traffic and cache miss and therefore reduces overall computation times.

The sparse solver is effective, although iterative solvers may be faster in certain applications. It is usually about 10-100 times faster than traditional direct solvers and requires storage of about 2-10 times than the iterative solvers.

The sparse solver can also go out-of-core in case the machine memory is not enough to hold the ordered matrix in the core. In this case, the memory must be explicitly assigned to the sparse solver. The required hard disk space will be automatically used whenever additional storage is needed. Several important observations regarding the sparse solver are listed here:

- If no memory is assigned to the sparse solver, it will try to allocate the memory that is required for an in-core solution. In case insufficient memory is available, the sparse solver prints information and ADINA-F terminates the execution. There is a possibility that the operating system goes to swap on disk when the required allocated memory is not sufficient. In this case, the sparse solver could be extremely slow (it is "out-of-core", but very inefficiently managed by the operating

system!). In iterative computing of the two-way fluid-structure interaction problems, the sparse solver frees the memory every time after the solid and fluid models are solved. There is a possibility that insufficient in-core memory is encountered during FSI iterations if the machine is shared by more than one job.

- If a specific memory is assigned to the sparse solver, it assumes an in-core procedure if the memory is sufficient. Otherwise, it will work on the hard disk, enforcing an out-of-core solution.
- The speed of the out-of-core sparse solver does not depend much on the memory assigned to it and is about two times slower than its in-core counterpart. Note that the original assembled matrix must always be in-core, even when the out-of-core sparse solver is used.

Using these facts, we may efficiently use the sparse solver as suggested below.

- Only when it is absolutely certain that there is sufficient memory on the machine for an in-core solution, it is not necessary to assign memory to the sparse solver.
- If it is doubtful whether the solution can be performed in-core, assign a reasonable amount of memory to the sparse solver.
- If the out-of-core solution becomes the only choice, assign just enough memory to ADINA-F and assign the rest of available memory to the sparse solver (see Chapter 13 for more details).
- In iterative computing of two-way fluid-structure coupling problems, always assign to the sparse solver the memory that is continuously available on that machine.
- If the operating system automatically goes to swapping in case of memory shortage, always assign memory to the sparse solver.

11.3.3 Iterative methods

There are five iterative solvers available in ADINA-F:

(1) Right preconditioned biconjugate gradient method (RPBCG)
(2) Right preconditioned generalized minimal residual method (RPGMRES)
(3) Algebraic multi-grid method (AMG) for system matrix

(4) Algebraic multi-grid method-1 (AMG1) for segregated method
(5) Algebraic multi-grid method-2 (AMG2) for segregated method

For illustration purpose, we will only present details of RPBCG in this section.

When a preconditioner is required, we apply a right preconditioning to the nonsymmetric linear system of equations. Instead of solving the original equations $Ax = b$, we solve the equivalent system

$$\tilde{A}\tilde{x} = b$$

where $\tilde{A} = AB^{-1}$ and $\tilde{x} = Bx$.

A typical preconditioner is obtained using the incomplete *LDU* decomposition of the matrix *A*. In this method, *B* is an approximation to the complete factorization of the matrix *A* obtained in the Gauss elimination method described in Section 11.1. The incomplete factorization is performed as follows. Consider only those locations where *A* has nonzero elements; at these locations, perform operations corresponding to the usual decomposition procedure using elements only from those locations where *A* has nonzero elements. The sparsity is then completely preserved in the resulting incomplete factors.

In this structure, the required storage is dramatically reduced. The matrix *B* only requires the same storage as the matrix *A*. With an additional locator array, the total storage requirement is about $5\alpha_m n_d N_d$ words in 32-bit machines, where the factor n_d varies from 1 to 7 corresponding to the number of solution variables and α_m varies from 8 for 2D 3-node triangular elements to 3125 for 3D 27-node brick elements, and N_d is the number of equations.

In the RPBCG method, we apply the biconjugate gradient method to solve the equation. The algorithm can be summarized as follows.

Algorithm RPBCG

$x = 0, r = \overline{r} = p = \overline{p} = b, \xi = (r, \overline{r})$

for $k = 1, 2, ...$

$z = B^{-1} p, \overline{z} = Az, \alpha = \xi / (\overline{z}, \overline{p})$

$x = x + \alpha z$

$r = r - \alpha \overline{z}$

if $(converged)$ *stop*

$z = A^T \overline{p}, \overline{z} = B^{-T} z$

$\overline{r} = \overline{r} - \alpha \overline{z}$

$\eta = (r, \overline{r}), \beta = \eta / \xi, \xi = \eta$

$p = r + \beta p$

$\overline{p} = \overline{r} + \beta \overline{p}$

The convergence criterion is

$$(1)\ \|r\| < \varepsilon_1 \text{ and } \|r\| / \|b\| < \varepsilon_2 \text{ or}$$
$$(2)\ (\|r\| < \varepsilon_3)$$

in which ε_1, ε_2 and ε_3 are predetermined tolerances. If the convergence criteria are not satisfied after *K* iterations, ADINA-F prints a warning message and terminates the current iterations in the solver. However, the program continues.

The convergence control in the algebraic multi-grid method is very much like in the RPBCG and RPGMRES methods. However, it requires slightly more storage in manipulating the restriction and prolongation operators in the coarsening and refining procedures respectively. The default values of the control parameters for iterative solvers are

$$K = 3000, \quad \varepsilon_1 = 10^{-6}, \quad \varepsilon_2 = 10^{-4}, \quad \varepsilon_3 = 10^{-8}$$

Although the convergence of the iterative methods has been proven in exact arithmetic for positive definite symmetric matrices, convergence is not guaranteed when nonsymmetric indefinite matrices (such as in fluid flow solutions) are considered. Nevertheless, numerical experiments show that iterative methods can converge fast if the conditioning of the matrix is fairly good and the number of the equations is not too large (typically in less than 1000 iterations if the number of equations is less than 200,000). For large equation systems, it is generally believed that the incomplete factorized preconditioner is farther from the original matrix and thus more iteration steps are required. Furthermore, the iterative methods require much less memory than direct solution methods.

The ADINA AMG solver is designed for large systems (more than 200,000 equations) and can be efficiently applied to most fluid flows. Currently, AMG solver cannot be used in direct computing of fluid-structure interactions, and to problems with sliding-mesh, gap and slip-wall boundary conditions. Its efficiency may be somewhat diminished in turbulence models or if more than one coordinate system is used.

ADINA AMG1 and AMG2 are specially designed in association with the segregated method. Their convergence control is introduced in the description of inner iteration of the segregated method.

Chapter 12 Other capabilities

12.1 Automatic time-stepping CFL option

It is well known that explicit methods are unstable if the time step length does not satisfy the CFL (Courant-Friedrichs-Lewy) condition

$$\Delta t \le \min\left\{\frac{\Delta x}{w + 2\mu/\rho\Delta x}\right\} \equiv \Delta t_c$$

where w is given by $|\Delta u| + c$ in high-speed compressible flows or the magnitude of the velocity $\|\mathbf{v}\|$ in other flows.

Implicit methods have no such restriction. However, the CFL condition is only a necessary condition obtained from a linear stability analysis. Fluid problems are always nonlinear and the nonlinear terms are dominant in many practical problems. Furthermore, experience has shown that a reasonable time step length may not only accelerate the global convergence in many cases, but also be crucial for stability.

We have developed the automatic time-stepping (CFL) option. In this option, a user specified CFL number, λ_{CFL}, plays a role of relaxation, rather than a corresponding real time. The dynamic terms in the governing equations are discretized

$$\frac{\partial\, {}^{t+\alpha\Delta t}f^k}{\partial t} = \frac{{}^{t+\alpha\Delta t}f^k - {}^{\Delta t}f^k}{\alpha\Delta t} + \frac{{}^{t+\alpha\Delta t}f^k - {}^{t+\alpha\Delta t}f^{k-1}}{\Delta\xi}$$

where $\Delta\xi$ is a fictitious time step length and the right superscript represents the iteration number in equilibrium iterations. The first term on the right hand side is the Euler α-method term that has been introduced in the formulations of fluid models. The second term is introduced as a fictitious time derivative that approaches zero when the iteration converges. It is also clear that the fictitious time step increases the diagonal dominance of the global matrix and thus improves its conditioning.

Focusing on the magnitude of the diagonal term (coefficient of the current variable ${}^{t+\alpha\Delta t}f^k$), we choose $\triangle\xi$ such that

$$\frac{1}{\alpha\Delta t}+\frac{1}{\Delta\xi}\geq\frac{1}{\lambda_{\text{CFL}}\Delta t_c}$$

By default, when this option is not applied, the CFL number is infinity. For most of the practical problems, $\lambda_{\text{CFL}}\sim 10^2$ to 10^7 can be used. For explicit time integration, $\lambda_{\text{CFL}}=0.8$ to 0.9 can be a good choice. In transient analyses, a proper use of the CFL number may improve the equilibrium iteration convergence at each time step. In steady-state analyses (where Δt is infinity), the convergence here means the final converged solution and the equilibrium iterations act as time steps (known as the time-marching method for steady-state solutions). In general, the larger the CFL number, the less stable the scheme or the more difficult the convergence, while the smaller the CFL number, the more stable the scheme and the less difficulty in convergence. However, the drawback of the smaller CFL number is a slower convergence. Considerably more iterations are required if the CFL number is too small. The magnitude 1 should be regarded as the lower limit of λ_{CFL} for implicit methods. An optimized CFL number can be obtained by numerical experiments and experiences. Usually one CFL number can be used for a group of models that have similar geometry, materials and boundary conditions.

Using this option and the control of loading increments in time steps can overcome many stability or convergence problems.

12.2 Automatic time-stepping ATS option

The automatic-time-stepping (ATS) option is used to obtain a converged solution when equilibrium iterations fail because too large a time step (giving too large load increments) is used.

In this option, the iterations in the implicit time integration are performed as described in related chapters. In case convergence is not reached, the program automatically subdivides the total load step increment into two equal sub-time-steps and tries to compute solutions at these time

steps. This procedure continues if the iterations still do not converge until a proper time step is found or the maximum subdivisions allowed have been reached.

In the first case, the program continues the next sub-time-steps until the user specified time step is reached. It may increase or decrease the sub-time-step again in order to achieve the best convergence rate.

The user controls the maximum number of subdivisions. When too many subdivisions are made, the program stops, indicating divergence of the computations at the current time step.

The converged solutions are saved only at the user specified time steps, regardless of the subdivisions that have been made.

The ATS option may help solution convergence in several ways as listed here.

- In transient analyses, the smaller time steps improve the matrix conditioning.
- When boundary conditions are time-dependent, smaller increments of time steps are usually equivalent to smaller increments of loads provided that the loads are monotonically increasing.

However, one must be cautious when using the ATS option. Since the time step length is successively cut into two equal sizes, the process can be very expensive if the original user-specified time step is far from the final one used. For example, 10 subdivisions will result in a final step of 0.001 from an original user-specified step length 1. In each subdivision, the maximum number of iterations will be performed to indicate an unsuccessful time step size so that the next subdivision is necessary.

12.3 Restart analysis

During the execution of a problem, the step-by-step solutions are automatically saved to a restart file every N time steps, where N is chosen by the user. Output to the restart file always overwrites the current contents of the file, i.e. the restart file only contains the solutions of the last saved time step. Therefore, if it is expected to use a restart file more than once, make sure to have a backup of that file since it will be overwritten in a restart run.

Now let us call the previous problem *A*. Using the solution of the problem *A* as an initial condition, another solution can be computed (call the problem *B*), starting at the last time the solution *A* is saved. The problem *B* is called a restart run. In subsequent restart runs, the geometry and most element data must be the same as those in the first run. However, the following changes are allowed:

- Time step control data, including number of time steps, time step lengths and time functions.
- Global control flags, including analysis type (steady-state analysis and transient analysis), CFL option, ATS option and their associated parameters.
- The parameters associated with the boundary conditions. For example, the values of prescribed conditions, distributed loads, parameters in special conditions, etc.
- The parameters associated with the material models. For example, the viscosity, the heat conductivity, etc.

Restart analyses can be used for moving mesh problems, including particularly fluid-structure interactions.

12.4 Mapping solution

Upon the request of the user, a mapping file can be created that contains the problem solutions in a neutral format. The solution includes the independent nodal solutions and element connectivity. The solutions are saved at each solution time whenever the porthole file is saved. The mapping file can be in a binary or an ASCII format.

The mapping file can be used to transfer problem solutions from one program data set to another. For example, the ADINA-F generated mapping file can be directly loaded through the AUI and the temperature solutions in the file are then transferred to nodes of an ADINA solid model for the purpose of a thermal stress analysis. Of course, the overall geometric domains in ADINA-F solid element groups and the ADINA solid model must be coincident.

The mapping files can also be used to transfer the solutions between two problems of the same program. For example, the solution of a problem in ADINA-F can be transferred (interpolated) to another problem in

ADINA-F to be used as initial conditions. Unlike restart analyses, the two problems can be completely different except the coincident domains.

12.5 Skew system

Skew systems are right-handed orthogonal systems that have arbitrary orientations with respect to the global Cartesian coordinate system. They are applicable in ADINA-F for ease of modeling.

The a-b-c axes of a skew system are defined using the input of direction cosines, Euler angles or axes.

Skew systems can be assigned to any nodal point or through the associated geometry in AUI. Once a node is associated with a skew system, the velocities and loads are referred to that skew system.

In many situations, conditions in skew systems can be reached by other options. For example, on a slip wall where only the normal velocity is zero, one could apply a skew system to its normal direction and then fix the velocity in that direction. When the wall geometry is highly irregular, the application of skew systems becomes not only tedious, but also almost impossible. In this case, simply applying the special boundary condition "slip-wall" is a much better choice.

In another example shown in the following figure, the velocity shall be prescribed in the radial direction. One way, of course, is to apply skew systems along the inlet line and then apply the velocity in the radial direction. However, a simpler method is to use a proper spatial function.

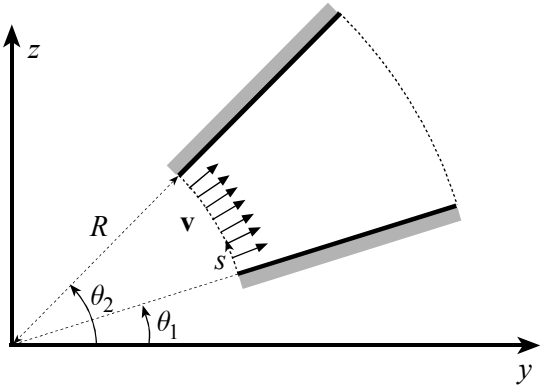


Figure 12.1 Prescribe a radial velocity

For example, we can prescribe the inlet velocity as

$$v = \overline{v} f(t) g_2(s)$$
$$w = \overline{v} f(t) g_3(s)$$

where $f(t)$ is the associated time function and the $g_i(s)$ are spatial functions which are, in this case,

$$g_2(s) = R\cos\left(\frac{s}{R} + \theta_1\right)$$
$$g_3(s) = R\sin\left(\frac{s}{R} + \theta_1\right)$$

12.6 Constraint conditions

In some analyses, it is necessary to impose constraints between solution variables. For example, a periodic boundary condition on solution variables defined on a part of the boundary is to be identical to the boundary condition defined on another part of the boundary.

In ADINA-F, the following conditions can be specified

$$f_k = \sum_{j=1}^{N_k} \beta_{kj} f_j$$

where f_k is a nodal solution variable such as velocity, pressure, temperature, etc., β_{kj} are multiplying factors and N_k is the number of independent variables used. The solution variables on the right-hand side are called master degrees of freedom or master solution variables, while the one on the left-hand side is called the slave degree of freedom or slave solution variable of these masters. The constraints can be explicitly specified for any of the nodal variables.

Fluid flow problems are usually highly nonlinear. This is particularly true in compressible flows. Special caution must be made when using

constraint equations. Constraining a single solution variable (say only pressure) is generally not a good choice. When using constraint equations, it is necessary to focus on the condition of entire geometric entities rather than considering only individual solution variables. For example, when considering a period boundary condition for surface 1 and surface 2, *all* solution variables of surface 2 should be constrained to the corresponding solution variables defined on surface 1.

12.7 Conjugate heat transfer problems

ADINA-F can be used to analyze problems of heat transfer in a domain that includes both fluids and solids, the so-called conjugate heat transfer problem. The heat transfer in solids occurs only by conduction. The heat transfer in fluids consists of conduction, convection and specular radiation, if specified.

The solid part can be defined through the definition of solid element groups (a special type of fluid element group). Defining a solid element group is the same as defining a fluid element group except that the "solid condition" is explicitly specified. The velocity degrees of freedom on nodes are then automatically deleted, including the interfaces with fluid element groups. The pressure degrees of freedom are also deleted except for the nodes shared with fluid element groups (see the illustration in the following figure).

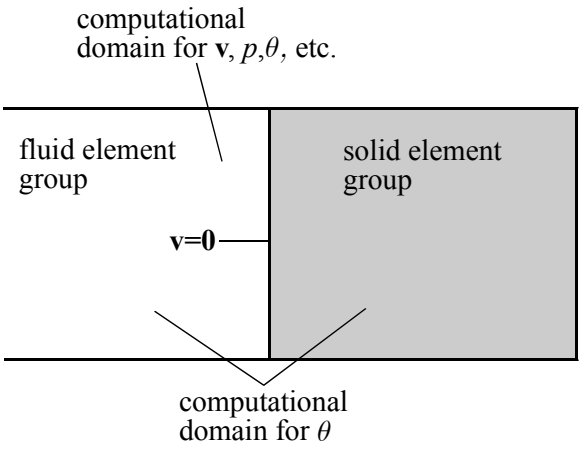


Figure 12.2 Sketch of conjugate heat transfer

The computational domain for the continuity equation and the momentum equations consists of all fluid element groups. Since the velocity degrees of freedom have been deleted on the interfaces between fluid and solid groups, a no-slip wall condition is implicitly applied. On the other hand, since the computational domain for the energy equation consists of the fluid and solid groups, the fluid-solid interface is not a boundary and therefore, no condition is required. The temperature solution is continuous across the interface.

12.8 Element birth-death option

The element birth and death option is available in ADINA-F. This option enables the modeling of changes in the computational domain. For example, an increased domain can be modeled by the element birth option, a reduced domain can be modeled by the element death option and an increased domain followed by a reduced domain can be modeled by the element birth-then-death option.

The boundary conditions must be applied to the domain that covers all possible live elements.

If the element birth option is used, the element is added to the total system of finite elements at the time of birth and remains active for all times thereafter. The boundary conditions that are linked to the element will also be alive.

If the element death option is used, the element is taken out of the total system of finite elements at the time of death and remains inactive for all times thereafter. The boundary conditions that are linked to the element will also be inactive.

If the element birth-then-death option is 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 element is then taken out of the total system of finite elements and remains inactive for all times thereafter.

12.9 Pressure datum

A pressure datum can be specified at a geometric point or node. Once this feature is used, ADINA-F shifts all of the pressure solutions by a pressure increment Δp (positive of negative)

$$p' = p + \Delta p$$

where, the increment pressure can be either one of

(1) $\Delta p = p_d(t) - p_i$

(2) $\Delta p = p_d(t)$

in which, i is the nodal label where the pressure datum $p_d(t)$ is specified. In the first case, the resultant pressure p' is equal to the specified pressure datum at the specific location. In the second case, it always adds the specified pressure datum.

This procedure of shifting the pressure is only performed in the output phase of the program. In particular, all stresses and forces that are printed, saved and exported are calculated using the shifted pressure (including the stress applied to structures in FSI problems). After output, the pressure solution is shifted back to the original solutions.

The pressure datum feature is useful in fluid-structure interaction analysis as shown in Fig.12.3. In the physical model (a), the normal-traction at the outlet could be a very large number. Since the pressure solution in incompressible flows can be shifted by a constant, such a big number can be avoided in the computation by specifying a zero normal-traction as shown in (b). However, the total stress acting on the structure must include the pressure that is omitted in the fluid model solution. This is accomplished by simply adding the omitted pressure back into the fluid stress calculated as shown in (c). Note that the pressure datum can be negative too. In this case, it is subtracted from the computed pressure solution that is applied to the structures.

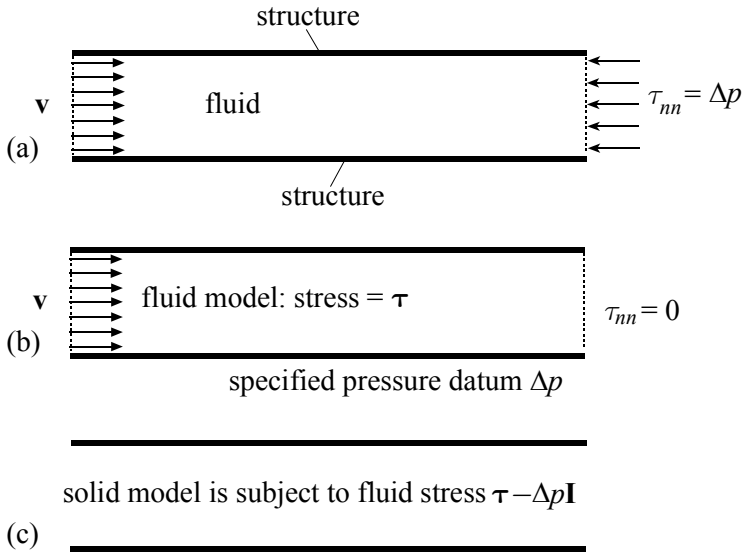


Figure 12.3 Application of pressure-datum feature to fluid-structure interaction problems.

12.10 ALE formulation and leader-follower option

The moving mesh conditions are encountered if any boundary condition is a moving boundary condition. These conditions are given by moving walls, free surfaces, fluid-fluid interfaces and, of course, most importantly the fluid-structure interfaces. The discussion in this section covers all types of moving mesh problems.

Physically, moving boundary conditions prescribe the boundary nodal positions. For example, the displacements of nodes on FSI boundaries are determined by the structural displacements; the displacements of nodes on moving walls are determined by the time functions; the displacement of nodes on free surfaces is determined by the free surface equations; and so on. However, the boundary nodes can be moved along the geometric boundary that changes position. The interior nodal movements are quite arbitrary but need to be controlled to keep a good mesh quality. Of course, there is no unique solution to establish a good mesh.

12.10.1 ALE formulation

In order to automatically determine the displacements of those nodes that can arbitrarily be moved, we currently solve the Laplace equation

$$\nabla^2 \Delta \mathbf{d} = 0 \qquad (1.15)$$

where $\Delta \mathbf{d}$ is the increment of the displacement. The latest displacement is then updated by adding the incremental solutions. A finite element method is applied to solve equation (1.15) based on either the latest nodal positions or the initial nodal positions.

However, Laplacian solutions cannot guarantee a valid mesh even in some simple domains. The following figure shows such an example.

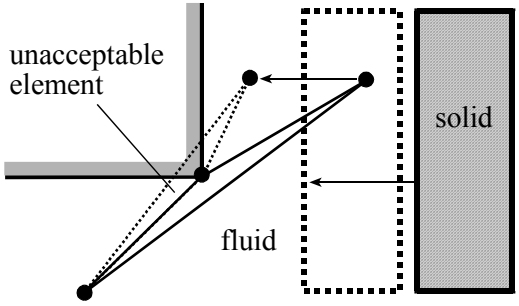


Figure 12.4 A typical case of reaching an unacceptable element.

When the fluid domain becomes more complicated, the situation can become worse. From a mathematical point of view, the closer the domain is to a convex domain, the greater the chance that the updated mesh is good. Hence, it is a good strategy to divide the computational domain into a few convex-like sub-domains and then apply the Laplace operator to these sub-domains. A revised element would then be perfectly fine, as shown in the following figure.

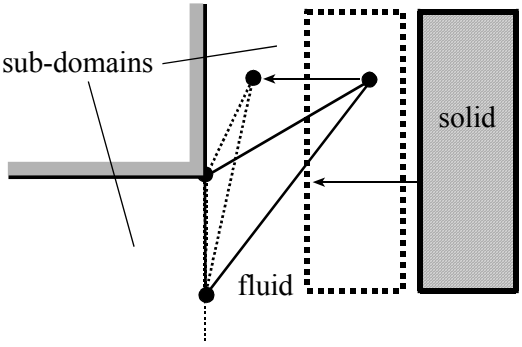


Figure 12.5 Valid mesh with a proper subdivision of the domain

Our general approach to reach a better solution of arbitrarily moving meshes is based on the following steps:

1) Determine the displacements of the boundary nodes from the physical conditions. These nodes are on fluid-structure interfaces, free surfaces, phase-change interfaces and fluid-fluid interfaces. They may also include some nodes controlled by users (see the Leader-follower option described in the next section).
2) Solve the Laplace equations on all lines that have been used to generate fluid elements. The displacements obtained in step 1 are used as prescribed conditions in this step. If the displacements of any end nodal points of the lines have not been prescribed from the physical moving conditions, they are set to zero. These end point conditions provide Dirichlet boundary conditions for the equations on lines. Note that these lines may include those that are actually inside the computational domain.
3) Solve the Laplace equations on all surfaces that have been used to generate fluid elements. The displacements obtained in step 2 are used as Dirichlet boundary conditions for these equations. Since lines enclose each surface, the surface equations are well defined.
4) Finally, if the problem is a three-dimensional problem, solve the Laplace equations on all volumes, with the Dirichlet boundary conditions obtained in step 3.

The program automatically performs these steps. However, the lines, surfaces and volumes are implicitly determined during the procedure of constructing elements. Therefore, the final mesh quality actually depends on how the computational domain is subdivided while the elements are generated.

12.10.2 Leader-follower option

In the ALE formulation, large displacements can be encountered. The leader-follower option is designed to help in controlling the arbitrarily moving mesh.

A leader and a follower form a pair. A leader is a point that is located on a physically moving boundary (fluid-structure interface, moving wall, free surface, etc.). A follower is a point whose movement is determined by

its leader using the method selected by the user. The motion of the moving boundary determines the displacement of the leader. On the other hand, any point that is not physically moving can be moved in an arbitrary manner. Defining a leader-following pair, you can control a point by forcing it to follow the leader. This procedure is equivalent to having some nodal displacements prescribed, of course, in the special manner that is described here.

There are three types of leader-follower options:

(1) Parallel option — when a follower is not on the boundary of the computational domain, its motion is completely determined by its leader

$$\Delta \mathbf{d}^f = \lambda \Delta \mathbf{d}^l$$

where the superscripts indicate the leader and follower nodes and λ is a factor for the user (the default is one). However, boundary followers must always stay on the boundary while they follow their leaders. Their motion is obtained from

$$\Delta \mathbf{d}^f = \lambda \Delta \mathbf{d}^l \bullet (\mathbf{I} - \mathbf{nn})$$

where $\mathbf{n}$ is the locally normal direction of the boundary.

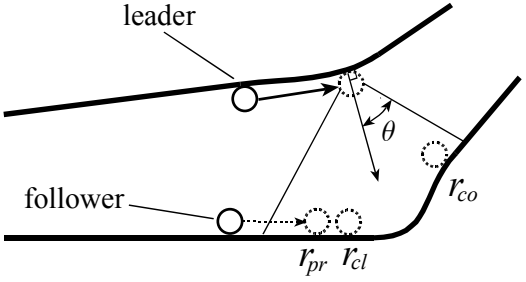


Figure 12.6 Definition of the leader-follower options

(2) Closest option — the boundary follower is located on the wall or boundary geometry at the "locally" closest location to the leader anywhere within the domain. In this option, the follower firstly moves as in the parallel option and, then, adjusts locally until it reaches the minimal distance.

(3) Cone option (for FCBI-C elements only) — the follower is located on the wall or boundary geometry at the locally closest location to the leader within a conical search domain. The leader associated with its normal to the boundary defines the apex of the conical domain and the cone angle θ is a user-specified variable.

In the illustrative figure, r_{pr}, r_{cl} and r_{co} represent follower positions with the options parallel, closest and cone respectively.

The leader-follower option is a very useful option in controlling the mesh quality in moving boundary problems. Some general guidelines for this option are described as follows:

- Leader and follower points are assigned to geometric points. The program will then transfer the condition to their closest nodes.
- The leader points must be on physically moving boundaries.
- As a general rule, the follower points must not be on physically moving boundaries. However, as an exception, they can be assigned to fluid-structure interfaces or moving walls. In this case, care must be exercised regarding the consistency of the physical and arbitrary motions.
- A follower cannot be assigned to be a leader.
- Closest and Cone options are proven useful associated with gap conditions. *Caution*: Avoid using these options if there are multiple closest points. In this case, the program will pick up one randomly as the follower's location point. This procedure may result in oscillation during iteration or even a diverged solution.

Recall that nodal displacements of lines are determined by the displacements of its end-points in the automatic procedure; points of leader-followers are sufficient to control line movements and, accordingly, to control surface and volume movements.

12.11 Electro-static and steady current conduction analyses

This capability allows the user to solve problems where volumetric heating of materials occur due to the existence of an electric current and problems where a electrostatic force is important. For incompressible, slightly compressible and low-speed compressible flows, we add the Joule heat generation term to the energy equation,

$$\rho C_v \frac{\partial \theta}{\partial t} + ... = ... + \tfrac{1}{\sigma}\mathbf{J}\bullet\mathbf{J}$$

and add the body force source term to the momentum equations,

$$\rho \frac{\partial \mathbf{v}}{\partial t} + ... = ... + \nabla\bullet\left(\mathbf{DE}\right) - \tfrac{1}{2}\nabla\left(\mathbf{D}\bullet\mathbf{E}\right)$$

where $\mathbf{J} = \mathbf{J}_\sigma + \mathbf{J}_0$ is the total current density, $\mathbf{J}_\sigma = \sigma\mathbf{E}$ is the unknown current density(or induced eddy current density), $\mathbf{D} = \varepsilon\mathbf{E}$ is the electric flux density, $\mathbf{E} = -\nabla\varphi$ is the electric field intensity, $\mathbf{J}_0$ is applied source current density, φ is the electrical potential, σ is the electrical conductivity and ε is the electric permittivity.

There are two different models available to obtain the electrical potential, namely electro-static analysis and steady current conduction analysis. The governing equations for the two models are

$$\nabla\bullet\left(\varepsilon\nabla\varphi\right) = -\rho_e$$
$$\nabla\bullet\left(\sigma\nabla\varphi\right) = \nabla\bullet\mathbf{J}_0$$

where ρ_e is electric charge density.

The electric permittivity ε and the electrical conductivity σ can be constant, time-dependent or temperature-dependent.

12.11.1 Discretization and method of solution

The variational form of Laplace's equation is, for example,

$$\int_V \left(\sigma \nabla \varphi \bullet \nabla h^{\varphi}\right) dV = -\oint h^{\varphi} \mathbf{J} \bullet d\boldsymbol{S}$$

The strategy of solution is as follows: Laplace's equation is solved for the electric potential, φ, at each time step. The Joule heat generation term, $\sigma \left\| \nabla \varphi \right\|^2$, is then calculated and added to the right-hand side of the energy equation.

12.11.2 Boundary conditions

12.11.2.1 Prescribed electric potential

In this condition, a time-dependent electric potential can be directly prescribed at the boundaries,

$$\varphi = \overline{\varphi}(t)$$

12.11.2.2 Distributed current density and electric flux density loads

When steady current conduction model is adopted, the current density condition can be active. With this condition, user specifies the normal component of the unknown current density, $\overline{J}_n(t) = \mathbf{n} \bullet \mathbf{J}_{\sigma}$.

When an electrostatic model is adopted, the electric flux density condition can be active. With this condition, the user specifies the normal component of the electric flux density, $\overline{J}_n(t) = \mathbf{n} \bullet \mathbf{D}$.

12.11.2.3 electric charge density and source current density

The field electric charge density or source current density can be applied through user-supplied subroutine ESUSRS. The model type (electrostatic or steady current), problem heading, element group ID, element ID, current time, spatial coordinate are passed into the subroutine. User writes its own program to calculate the source term. When the model is electrostatic analysis, the electric charge density (scalar) is expected and, when the model is steady current analysis, the source current density (vector) is expected.

12.12 Phase change between liquid and vapor

ADINA-F can be used to analyze problems with phase change between liquid and vapor. There are two different options available for phase change. For the first option (general phase change model), the phase change interfaces are determined by the temperature distribution in the fluid. For the second option (cavitation model), the interfaces are determined by the pressure distribution in the fluid. This feature is available for incompressible flows, slightly compressible flows and low-speed compressible flows. Different flow types can be chosen for different phases. All elements that are currently used in ADINA-F can be used for this feature.

Once the phase-change option is used, the density is always treated as a variable even in incompressible flow models. The surface tension effect at the phase-change interface is not available in current models.

We use the so-called vapor fraction function f that is defined as the rate of the vapor volume and total volume. Any mixed property of a material parameter m will be calculated as

$$m = m_l + f\left(m_v - m_l\right)$$

where the subscripts l and v represent, respectively, the liquid phase and vapor phase.

12.12.1 General phase change model

In this model, the energy equation is modified to include the effect of latent heat (L):

$$\rho C_v \frac{\partial \theta}{\partial t} + \cdots = \cdots - \rho \frac{\partial fL}{\partial t} - \rho \mathbf{v} \bullet \nabla \left(fL\right)$$

First let's consider the phase change between liquid and vapor. Let θ_{lv} and θ_{vl} be evaporation temperature and condensation temperature respectively ($\theta_{lv} < \theta_{vl}$) and assume that the latent heat is absorbed when liquid changes to vapor. The vapor fraction f thus equals zero under evaporation temperature and equals one above condensation temperature. A linear function is used to approximate the function between the two temperature values. In order to obtain continuous derivatives, we also smooth the linear function near the evaporation temperature and condensation temperature within a region of length $\alpha\left(\theta_{vl} - \theta_{lv}\right)$.

To be more specific, we first define a normalized function as

$$\overline{f}\left(x,\alpha\right) = \begin{cases} 0 & x \le 0 \\ x^2/d & 0 < x \le \alpha \\ \left(2x\alpha - \alpha^2\right)/d & \alpha < x \le 1-\alpha \\ 1-\left(x-1\right)^2/d & 1-\alpha < x \le 1 \\ 1 & x > 1 \end{cases} \qquad \left(d = 2\alpha\left(1-\alpha\right)\right)$$

Currently, we choose $\alpha = 0.1$. In the general phase-change model, we use $f = \overline{f}\left(\tilde{\theta},\alpha\right)$, where $\tilde{\theta} = \left(\theta - \theta_{lv}\right)/\left(\theta_{vl} - \theta_{lv}\right)$.

For phase-change between solid and liquid, the energy equation above is still valid, but the above evaporation temperature θ_{lv} and the condensation temperature θ_{vl} are replaced with the melting temperature θ_{sl} and the solidification temperature θ_{ls}, respectively. To mimic solid regions, the velocities and/or turbulent variables should be suppressed in solid regions. This suppression is achieved by additional source terms added into the momentum equations and the turbulence equations.

For melting and solidification applications, the additional term $A\left(\mathbf{v} - \mathbf{v}_p\right)$ is added onto the right hand side of the momentum equation,

and $A\phi$ is added onto the right hand side of the turbulence equations. In these terms, $\mathbf{v}_p$ is the solid region pulling velocity, and ϕ represents k, ε, ω. The coefficient A is defined as

$$A = -C_{mushy}\frac{(1-f)^2}{\left(f^3+\varepsilon_{mushy}\right)}$$

where f is the liquid fraction, C_{mushy} and ε_{mushy} are user-defined parameters. Note: $C_{mushy} \geq 0$ and $\varepsilon_{mushy} \geq 0$. If C_{mushy} or ε_{mushy} is less than zero, the computations may be unstable. Generally the value of C_{mushy} is about $10^4 - 10^7$ times larger than any other terms in the momentum and turbulence equations to suppress velocities and turbulence variables in solid regions. ε_{mushy} is a small constant to avoid numerical instability as f is very small. Larger C_{mushy} and/or smaller ε_{mushy} is, harder the convergence is.

12.12.2 Cavitation model

In liquid flows containing micro-size gas bubbles, cavitation generally occurs when the pressure at some locations is below the vapor pressure of the liquid. Cavitation appears in a wide variety of engineering applications, such as nozzle, pumps, hydrofoils, propellers, and so on. In almost all engineering applications, cavitation is not a desirable phenomenon. It can cause structure damage, generate undesirable vibration and noise, and lower system performance efficiency.

The default cavitation model in ADINA-CFD is a simplified phase change model. The effect of latent heat is omitted. The phase change is assumed primarily caused by the pressure change. Let p_{lv} and p_{vl} be the evaporation pressure and the condensation pressure respectively ($p_{lv} > p_{vl}$), the vapor fraction function is evaluated by $f = \overline{f}\left(\tilde{p}, \alpha\right)$, where $\tilde{p} = \left(p_{lv} - p\right)/\left(p_{lv} - p_{vl}\right)$.

A more sophisticated cavitation model associated with the VOF method is also available (currently only for FCBI-C element). This model cannot be used for compressible flows and free surface flows. Non-condensable gas is considered in the ZGB model. Also note that the problem must be in transient.

To setup this model, one species of VOF must be activated. In the VOF method, the vapor volume fraction f_v and liquid volume fraction f_l are defined as

$$f_v = \frac{V_v}{V_v + V_l}, \quad f_l = 1 - f_v$$

where V_v and V_l are vapor volume and liquid volume respectively.

The mixture is a hypothetical fluid with variable density:

$$\rho_m = f_v \rho_v + f_l \rho_l$$

The mixture satisfies the continuity equation:

$$\frac{\partial \rho_m}{\partial t} + \nabla \bullet \left(\rho_m \mathbf{v} \right) = 0$$

The transport of equation of the liquid volume fraction is

$$\frac{\partial f_l}{\partial t} + \nabla \bullet \left(f_l \mathbf{v} \right) = -\frac{1}{\rho_l} \dot{m}$$

The mass transfer between the vapor and liquid is modeled through the mass creation rate $\dot{m}$.

In practical applications, the cavitation number is one of the most important dimensionless numbers used to describe cavitating flow structures. Its definition is

$$\sigma_{cav} = \frac{p_\infty - p_v}{\frac{1}{2}\rho_l v_\infty^2}$$

where p_∞ and v_∞ are the far-field pressure and velocity respectively, and p_v is the vapor pressure of the liquid.

ADINA-CFD provides two models to simulate $\dot{m}$: Kunz model and ZGB model.

12.12.2.1 Kunz model

Kunz et al. modeled the vapor-liquid mass transfer rate as below:

$$\dot{m} = \frac{C_{dest}\rho_v f_l \max\left(p_v - p, 0\right)}{\frac{1}{2}\rho_l v_\infty^2 t_\infty} - \frac{C_{prod}\rho_v f_l^2 f_v}{t_\infty}$$

where $t_\infty \left(= L/v_\infty\right)$ and L are the characteristic system scales of time and length respectively, and the C_{dest} and C_{prod} are two user-defined parameters.

12.12.2.2 Zwart, Gerber and Belamri (ZGB) model

ZGB model is given as below:

$$\dot{m} = \begin{cases} C_{dest}\rho_v \dfrac{3 f_{nuc} f_l}{R}\sqrt{\dfrac{2}{3}\dfrac{p_v - p}{\rho_l}} & \text{if } p \le p_v \\ -C_{prod}\rho_v \dfrac{3 f_v}{R}\sqrt{\dfrac{2}{3}\dfrac{p - p_v}{\rho_l}} & \text{if } p > p_v \end{cases}$$

where f_{nuc} is the nuclear site volume fraction, and R is the bubble radius. The suggested values for f_{nuc} and R are 0.0005 and 10^{-6}, respectively.

The coefficients C_{dest} and C_{prod} are flow-condition dependent parameters. The suggested values of C_{dest} and C_{prod} are 50 and 0.01 respectively. They are suitable for a variety of flows.

When a non-condensed gas is considered, the vapor volume fraction f_v is represented as

$$f_v = 1 - f_l - f_g$$

where f_g is the non-condensed gas volume fraction.

12.12.2.3 Turbulence viscosity modification

For cavitating flows, especially for the transitional flows from the sheet cavitation to the cloud cavitation, the compressibility of the mixture near the cavitation interfaces may need to be taken account in the turbulence model. For such cases, the mixed density viscosity ρ_m is further modified as

$$\overline{\rho}_m = \rho_v - \left(\frac{\rho_v - \rho_m}{\rho_v - \rho_l} \right)^n \left(\rho_v - \rho_l \right)$$

where the exponent constant n should be $n \geq 1$. For larger cavitation number applications, n can be set as 1. Its value increases as the cavitation number decreases and Reynolds number increases.

12.12.3 Phase-change model setup

To use this feature, you first declare that the element group is one with phase change. You then define and apply two sets of material data and additional control data. The flow types of the two phases can be different inside one element group. The phase-change temperatures (or the phase-change pressures), the latent heat and the type of phase (liquid or vapor) are defined in the control data.

We note that when the difference between the two phase-change temperatures (or the phase-change pressures) is very small, the convergence may be difficult. It is therefore important to specify proper control parameters.

12.13 VOF method

The VOF method is a surface-capturing method for solving flows possessing multiple immiscible fluids. In the VOF method, the fluids share momentum equations, and each of the fluids is convected through the domain by solving scalar transport equations. The surface tension and wall adhesion are simulated with a continuum surface force (CSF) model.

There are maximum 2 additional species that can be computed using the VOF method. This method is available for all types of elements in incompressible and slightly compressible flows. Currently it cannot be coupled with the phase-change feature.

12.13.1 Transport equations

The scalar transport equation used can be written as

$$\frac{\partial \phi_i}{\partial t} + \mathbf{v} \bullet \nabla \phi_i = 0 \qquad i = 1,..,n$$

where **v** is the velocity vector, ϕ_i is mass ratio of the *i*-th species, and *n* is total number of additional species (beside the primary fluid).

A finite volume method is used in discretizing the equation. A donor-acceptor scheme is used to evaluate the value of the mass ratio at the face of the control volume. In terms of accuracy, time step length must be sufficiently small (usually Courant number $v\Delta t/\Delta x \approx 0.1$).

12.13.2 Material

Any material parameter m in the mixed fluid domain is evaluated as an average among all the species,

$$m = \sum_{i=0}^{n} \phi_i m_i$$

We note that the mass-ratio for the primary fluid is $\phi_0 = 1 - \sum_{i=1}^{n} \phi_i$.

12.13.3 Surface tension

The surface tension and the wall adhesion are simulated with a continuum surface force (CSF) model that models the surface force as a body force. The body force per unit volume is then evaluated using

$$\mathbf{f}^{\mathbf{s}} = \sum_{i=0}^{n-1} \sum_{j=i+1}^{n} \sigma_{ij} \left(\phi_i \Gamma_j \nabla \phi_j + \phi_j \Gamma_i \nabla \phi_i \right)$$

where σ_{ij} is surface tension coefficient between species i and j, and $\Gamma_i = -\nabla \bullet \left(\nabla \phi_i / \left\| \nabla \phi_i \right\| \right)$.

12.13.4 VOF-wall angle condition

User can specify the angle conditions $\theta_{ij} \left(i < j \right)$ at the boundary, where θ_{ij} denotes the angle between the boundary outward normal and the normal of the interface (pointing from species j to the species i), as defined in the following figure.

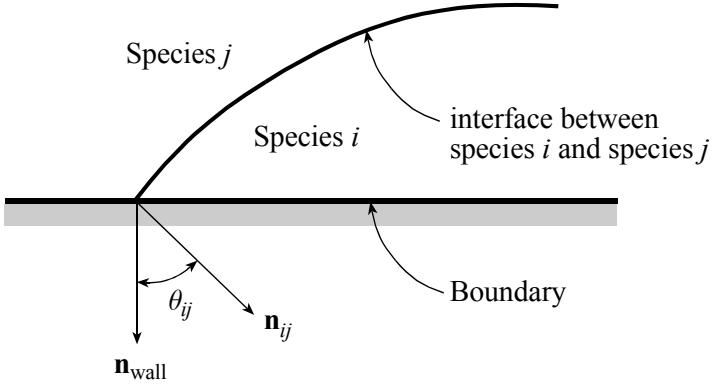


Figure 12.7 Definition of VOF-wall angle

12.13.5 Liquid-solid phase change for primary species

Phase change between liquid and solid can be considered for the primary species. Let θ_s and θ_l be the solidus temperature and liquidus temperature of the primary species ($\theta_s < \theta_l$) and assume that the latent heat is absorbed when solid changes to liquid. The volume fraction of liquid phase inside total volume of primary species $\tilde{f}_0$ thus equals zero under solidus temperature and equals one above liquidus temperature. A linear function is used to approximate the function between the two temperature values.

The latent heat effect is modeled as a source term adding to the right-hand side of the energy equation.

$$\rho C_v \frac{\partial \theta}{\partial t} + ... = ... - \rho_0 f_0 \left[\frac{\partial \left(\tilde{f}_0 L \right)}{\partial t} + \mathbf{v} \cdot \nabla \left(\tilde{f}_0 L \right) \right]$$

where ρ_0, f_0 and L are the density, volume fraction and latent heat of primary species. Simultaneously, the additional drag force related terms are added to the momentum equations and turbulence kinetic energy equation respectively

$$\rho \frac{\partial \mathbf{v}}{\partial t} + ... = ... - dC_{mush} \left(\mathbf{v} - \overline{\mathbf{v}}_s \right)$$

$$\rho \frac{\partial K}{\partial t} + ... = ... - dC_{mush} K$$

where,

$$d = \frac{\tilde{\tilde{f}}_0^2}{\left(1 - \tilde{\tilde{f}}_0 \right)^3 + \varepsilon}$$

C_{mush} is a mushy zone constant (of unit density/time), $\overline{\mathbf{v}}_s$ is a prescribed solid velocity, ε is a small constant introduced to avoid division by zero, $\tilde{\tilde{f}}_0 = \left(1 - \tilde{f}_0\right) f_0$ is the volume fraction of solid phase inside total volume (including all species). Notice that when the solid phase of primary species occupies the total volume, i.e. $\tilde{\tilde{f}}_0 = 1$, the right-hand side source terms of the momentum and turbulence kinetic energy equation dominate the equations. In other words, the equations change to

$$\mathbf{v} - \overline{\mathbf{v}}_s = 0, \quad K = 0$$

12.13.6 Loading of the species

The mass-ratios $\phi_i, \left(i = 1,..,n\right)$ can be prescribed at any lines, surfaces, volumes and nodes.

12.14 Rigid motion of element group

In certain applications, it is useful to specify mesh movement. This feature is particularly useful when associated with sliding-mesh boundary conditions. There are two types of rigid motions, translation and rotation, are available now. Under these conditions, all nodal positions in the specified element groups are determined.

The translation is defined as

$$\mathbf{x}\left(t\right) = \mathbf{x}^0 + \Delta\overline{\mathbf{x}}\left(t\right)$$

where $\mathbf{x}^0$ is initial nodal position and $\Delta\overline{\mathbf{x}}\left(t\right)$ is the specified time-dependent displacement.

In the condition of rigid rotation, the center of rotation $\overline{\mathbf{x}}_0$, rotation axis $\overline{\mathbf{e}}_3(t)$ and angular velocity $\overline{\Omega}(t)$ are specified. The nodal position is then determined by

$$\mathbf{x}(t) = \overline{\mathbf{x}}_0 + J^T R(\varphi) J\left(\mathbf{x}^0 - \overline{\mathbf{x}}_0\right)$$

where, J is the Jacobian that transforms global Cartesian system into a local system $\left(\mathbf{e}_1(t), \mathbf{e}_2(t), \overline{\mathbf{e}}_3(t)\right)$, and $R(\varphi)$ is the rotation matrix on plane $\left(\mathbf{e}_1(t), \mathbf{e}_2(t)\right)$. The rotated angle is computed from $\varphi = \int_0^t \overline{\Omega}(t)\, dt$.

12.15 Steered adaptive meshing

12.15.1 Introduction

The steered adaptive meshing (SAM) capability is applicable for CFD and FSI analyses. The capability is particularly useful for steady-state analysis.

In a SAM procedure, the analysis proceeds in the usual time stepping solution and invokes the adaptive meshing step in the following cases.

- Iteration convergence is not reached in a time step. This is frequently due to a poor mesh, which is therefore repaired.
- Mesh crashes during the solution iteration (typically in FSI analysis). This adaptation is also called mesh repair.
- Once a solution over the complete time span has been reached, additional adaptive steps are prescribed to improve the solution accuracy. This is referred to as mesh adjustment.

12.15.1.1 Adaptive mesh repair

Suppose that a CFD or FSI analysis is required for a number of time steps from time $t = t_0$ to time $t = t_{end}$, where $t_0 < t_{end}$, and suppose that the usual

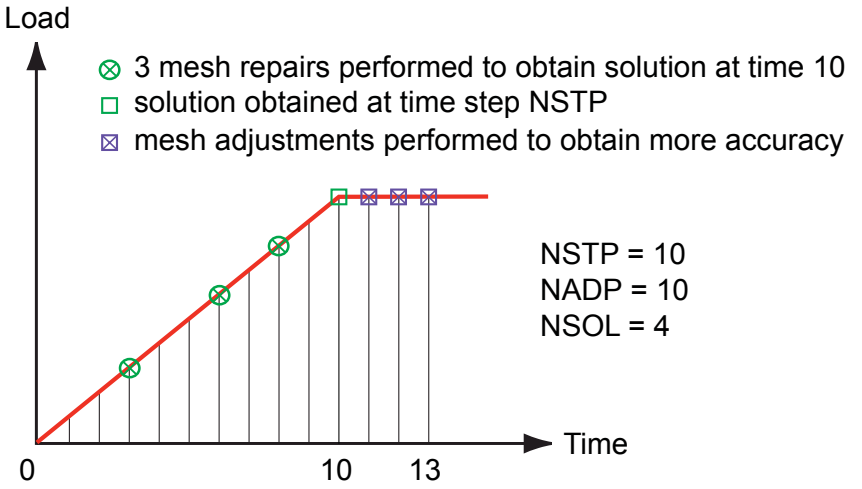
step-by-step procedure is performed, but the program does not converge at a time step $t = t_c$. Then with SAM and the prescribed remeshing criterion, a new mesh is automatically generated from a mesh at a previous time step t_p ($t_0 < t_p < t_c$). Thereafter the solution is automatically continued with the new mesh. If necessary, this procedure of mesh repair is repeated at further time steps until the computation finishes at the expected time $t = t_{end}$.

12.15.1.2 Adaptive mesh adjustment

Suppose that a CFD or FSI analysis has been performed and the computation has been completed for the total number of time steps. However, the analyst judges that the result needs to be improved with a better mesh quality. SAM can be set up for such mesh adaptation to achieve solution accuracy. Let

NSTP	=	total number of time steps in original model
NADP	=	maximum number of adaptive steps (steps with remeshing) that can be performed
(NSOL-1)	=	number of additional solution steps after the solution with NSTP steps has been performed. NSOL is typically 1 (default), so that no further solution steps are performed after the NSTP number of steps. But NSOL can be greater than 1, if the analyst requires higher accuracy in the solution. Then, in each such step a new mesh will be spanned to reach that accuracy. Note that if during the NSTP solutions, there have been M adaptive steps, then (NADP-M) must not be less than (NSOL-1).

Here we give an example.



In this example, the number of initial total time steps NSTP=10. At steps 3, 6, and 8, remeshing has been performed to reach a solution at time 10; hence M=3.

Here, NADP-M=7 and hence 3 = (NSOL-1) additional solution steps corresponding to times 11, 12, 13 are performed, each with a remeshing in an attempt to obtain the requested accuracy.

Note that the following steps are required in the ADINA User Interface (AUI) for additional adaptive steps to achieve better accuracy.

- In Meshing→Steered Adaptive Mesh→Control, user must set "Adaptive Timestep Sequence" to "**Appends** Original Timestep"

- User must also define a time step with name “ADAPTIVE” with a single step of size 1.0

The user can refer to Primer Problem 48 for more details on the use of additional adaptive steps for better accuracy in the solution.

12.15.2 General procedure

A CFD/FSI computation with SAM may have one or more mesh repair or mesh adaptation steps, all of which we refer to here as restarts. Each restart is referred to as a new model because it has a new mesh, and also its material properties, loads, initial and boundary conditions could be changed at restart times (if the manual mode is selected). The model at initial time $t = t_0$ is referred to as the first model, and the model at the first restart time as the second one, and so on.

The General procedure of both Manual mode and Automatic mode includes two steps: Model setup and Running the CFD/FSI model with SAM.

12.15.2.1 Model Setup

Structure Model Setup for FSI problems

As mentioned before, SAM does not affect the structural models in FSI problems. Therefore, there are no special restrictions on structural models regarding geometry, element type, etc. for FSI problems with SAM.

Fluid Model Setup for CFD/FSI problems

(1) Geometry and Mesh Preparations in Fluid Models

For CFD models and FSI fluid models, geometries have to be defined as bodies via ADINA-M in AUI, or imported from a Nastran file. The element type has to be 3-node triangular elements for 2D problems or 4-node tetrahedral elements for 3D problems because SAM only generates free-formed meshes with these element types. These basic requirements of SAM are listed in the following table for clarity:

Table 12. 1 Geometry generation tools and mesh element type requirements for SAM

Geometry generation tools	ADINA-M or Nastran
Mesh element type	3-node triangular elements (2D) 4-node tetrahedral elements (3D)

Note: internal and external boundaries in the first model will not be changed when new meshes are generated via SAM. That is, even though the nodal points on geometry boundaries can be re-distributed, the boundaries cannot intercept with each other. Any internal and/or external boundary interception will make the SAM fail.

(2) Other Settings in Fluid Models

Other fluid model settings of CFD/FSI with SAM, such as initial conditions, boundary conditions, materials and solution process parameters, are the exactly same as those of general CFD problems.

(3) SAM Setup

In AUI, SAM can be accessed in *Meshing* → *Steered Adaptive Meshing*. There are three dialog boxes for SAM: control, criterion and meshing. In SAM control dialog box, Manual mode or Automatic mode can be selected; restart time, at which SAM will generate a new mesh, can be specified; CFD restart file (*.res) and adaptive command file (*.adp) can be chosen; interpolation schemes of boundary geometry segments and mapping schemes of boundary conditions and initial conditions can be specified.

In SAM criterion dialog box, six criteria are available: Element Quality, Element Size, Variable Gradient, Sphere Region, Brick Region and Direct List. Each criterion can be used individually, or they can be used in a combined way. See detail description for these criteria in the sections of criteria for selection of preferable element size and combination of criteria.

If Manual Mode is used, users will manually give all SAM control information and criteria. Users may also change initial conditions, boundary conditions, and solution process parameters for restart models. In other words, users will manually generate the second model, the third model and so on by themselves.

If Automatic Mode is used, users will prepare the first model, specify criteria for re-meshing, and use AUI to generate an adaptive file (*.adp). The adaptive file includes control information used to create a new fluid model with a new mesh. For FSI problems, users will generate two adaptive files via AUI: one is for the fluid model and the other for the structural model.

More details of Manual Mode and Automatic Mode are given in the next section.

12.15.2.2. Running CFD/FSI model with SAM

Manual Mode

Step 1. For CFD problems, in the first fluid model, Manual mode is selected. Run the first CFD model as usual. Since the SAM is turned on in the first model, ADINA will save the adaptive file automatically. For FSI problems, besides in fluid model, SAM is turned on and Manual Mode is also selected in the first structural model. Run the first FSI model as usual. AUI will generate two adaptive files; one is for the fluid model and the other for the structural model. The structural model solutions will be saved at the same times at which the fluid model solutions are saved;

Step 2. Prepare restart model with adaptive mesh. At first, in SAM control dialog boxes, restart time, fluid model restart file, fluid model adaptive file can be specified, and then criteria for re-meshing in SAM criterion dialog box. At the end, a new mesh is generated via SAM meshing dialog box;

Step 3. Specify time step for the restart run;

Step 4. Save restart data file (*.dat);

Step 5. For FSI problems, use AUI to open the previous structural model at restart time. In SAM dialog box, select Manual Mode, specify restart time and specify restart file of the fluid model. For CFD problems, this step is skipped;

Step 6. Generate new structural model data file. For CFD problems, this step is skipped;

Step 7. Run CFD or FSI. For FSI problems, AUI will ask users to copy the structure restart file at restart time.

Steps (2-7) can be repeated until final solution is obtained.

Automatic Mode

Setting up the model

Step 1. Generate solid model (if it is an FSI problem) as usual in AUI-ADINA-STRUCTURE;

Step 2. Still in AUI-ADINA-STRUCTURE, open SAM dialog box, select SAM Automatic mode; give restart time (if the latest mesh is used, give -1); give corresponding CFD restart file if it is not a new run, otherwise leave it as blank;

Step 3. Save solid model;

For CFD problems, step 1-3 can be skipped.

Step 4. In AUI, select ADINA-CFD (with ADINA-Structure if it is a FSI problem);

Step 5. Generate fluid model as usual (see other CFD cases in primer). Notice that geometries can only be generated by using ADINA-M, and the mesh must be free mesh with triangles (2D) or tetrahedral elements (3D);

Step 6. Still in AUI-ADINA-CFD, open the SAM control dialog box and go to Time Step (*meshing* → *SAM* → *control* → *time step*) and define the time steps for the computation.

Note that the computation will be finished when all the time steps defined here are completed, regardless of the number of re-meshing processes performed during the run. In automatic SAM, the user can add more time steps to the original model in order to get more time solutions. This is described in the next step.

Step 7. In the SAM control dialog box (*meshing* → *SAM* → *control*), select SAM Automatic model; enter the restart time (if the latest mesh is used, specify "-1"); then define the SAM criteria in the criterion dialog box.

In automatic SAM, the re-meshing process will automatically take place when the computed solution diverges or fails due to mesh overlapping. A user can also choose to apply re-meshing at specific times with the "*Solution Times to Perform Adaptive Meshing*" option settings by filling a Solution Times table with a list of times that re-meshing should be performed. As well, in the "*Adaptive Timestep Sequence*" option settings, the user can choose to keep the number of time steps in the original model by choosing "Overwrite", or to retain a copy of the original time steps at every adaptive re-mesh run by choosing "Append".

Step 8. Save the fluid model.

Running the Automatic SAM

To run automatic SAM, with/without FSI, three options in the dialog box need to be filled:

(a) *Run Analysis from Adaptive Step*: If this is the new run for a model, use the default 0; otherwise, give the number to start the computation with.

(b) *Maximum Number of Adaptive Steps*: This is the maximum number of re-meshing processes that can be performed; if this maximum number is reached, no more re-meshing will take place.

(c) *Number of Solution Runs*: This is the maximum number of solution runs that can be performed. In the model, the completion of the all time steps in *.dat file is considered as one run. Note that the number of time steps in the *.dat file is not necessarily equal to the total number of time steps in the original model, because this number is also controlled by the "*Solution Times to Perform Adaptive Meshing*" option setting described above. The program will proceed to re-meshing and the next run until the maximum number of runs has been reached.

12.15.3 Criteria for selection of preferable element sizes

The novelty in our adaptation scheme is how we determine preferable element sizes, h_{ep}, using various criteria, denoted as $C(F_e)$, where it may depend on a flow-solution-variable F_e. Note that the criterion can be calculated at a time t_c (called criterion time) that could be earlier than or the same as the restart time t_s. The criteria can be applied to the whole fluid model or only to certain element groups.

The simplest criterion is to directly specify element size (a length scale) in a defined region Ω (e.g., a sphere or a brick)

$$C(F_e): \qquad h_{ep} = \overline{h}_{ep} \qquad e \in \Omega$$

Another type of criteria is based on the mesh deformations $r(F_e)$: relative element size variation or relative element quality variation

$$C(F_e): \qquad h_{ep} = \overline{h}_{ep} \qquad r(F_e) \notin [r_{\min}, r_{\max}]$$

where $r_{\min}$ and $r_{\max}$ are, respectively, the acceptable minimum and maximum variations.

A more sophisticated type of criteria is based on the well-known general fact that, whatever field is to be predicted accurately, the element size times the gradient of that field should be about constant over the fluid region

$$C(F_e): \qquad h_{ep}\|F_e\| = c$$

where c is a constant, and F_e here represents pressure gradient, vorticity, etc. The constant is determined by

$$c = \lambda_r \frac{1}{N_e} \sum_e h_e \|F_e\|$$

where λ_r is a ratio factor, h_e is the local element size at criterion time, and N_e is the total number of elements in that mesh.

In practice, elements of too large or too small sizes must be prevented. The calculated preferable element sizes are then modified by

$$h_{ep} = \frac{c}{\max\left\{\min\left\{\|F_e\|, \frac{c}{h_{\min}}\right\}, \frac{c}{h_{\max}}\right\}}$$

where $h_{\min}$ and $h_{\max}$ are, respectively, the minimum and maximum element sizes allowed in the mesh.

The choice of F_e depends on the type of problem. For example, the pressure-gradient criterion $C(\nabla p_e)$ may be used for having a sufficient number of elements in areas where large pressure gradients are present. The vorticity criterion $C(\nabla \times \mathbf{v})$ may be used to improve the solution accuracy in boundary layers. In fact, the gradient of any other anticipated solution variable could be used.

Based on a criterion, the program produces a "to-be-repaired-element set" E of elements e associated with preferable element sizes h_{ep}

$$C(F_e) \rightarrow E\{e \mid h_{ep}\}$$

12.15.4 Combination of criteria

Usually, one criterion is not sufficient to produce effective meshes. Therefore we allow certain operations on criteria to result in a combined set of elements to be repaired. The procedure is performed as follows

$$E = \varnothing$$
$$for\ i = 1,2,... \quad \{E =: operation\ of\ E\ and\ E_i\}$$

These operations are summarized in the following table

Operation	Definition	Note	
Append	$E \cup (E_i \setminus E)$	Append E_i to E	
Replace	$E_i \cup (E \setminus E_i)$	Append E to E_i	
Subtract	$E \setminus E_i$	Subtract E_i from E	
Smaller	$(E \setminus E_i)$	Join E and E_i	$h_{ep} =: \min\{h_{ep}, h_{epi}\}$
Larger	$\cup (E_i \setminus E)$	and update h_{ep}	$h_{ep} =: \max\{h_{ep}, h_{epi}\}$
Average	$\cup (E \cap E_i)$	if $e \in E \cap E_i$	$h_{ep} =: \frac{1}{2}(h_{ep} + h_{epi})$

12.15.5 Smoothing technique

The preferable element sizes may be smoothed a few times using the iteration procedure

$$h_{ep}^{k+1} = \frac{\sum_{nb} h_e h_{ep}^k}{\sum_{nb} h_e}; \qquad k = 1,2,...$$

where as indicated by "*nb*" the sum is carried out over the neighboring elements.

12.15.6 Control on the maximum number of elements

The above element criteria basically re-scale element sizes to obtain a mesh in which the selected gradients times the element sizes are about constant and the geometry and boundary conditions are properly represented. In practice, the model size must be limited, that is, there must be an upper limit on the maximum number of elements allowed. When the total number of preferable elements N_{ep} is larger than the maximum number of elements allowed $N_{\max}$, we can increase the geometric size of all elements by a factor λ, that is, we change h_{ep} to λh_{ep}, with

$$\lambda = \max\left\{\left(\frac{N_{ep}}{N_{\max}}\right)^{\frac{1}{\sigma}}, 1\right\}$$

where σ is the space dimension and

$$N_{ep} = \sum\nolimits_e \frac{V_e}{V_{ep}}$$

with V_e the current element volume and V_{ep} the volume of the elements computed using the preferable element size.

Of course, depending on the meshing algorithm used, the final number of elements obtained after this adaptation could still be slightly larger or smaller than $N_{\max}$. The purpose of this scheme is to apply a limit on the size of the model.

12.15.7 Comments and limitations

In principle, the adaptive-mesh option can be used for solving transient problems. In practice, however, the error accumulation due to the changes

in meshes over many time steps could be too large. The difficulty is that any error introduced at a particular time may not be "iterated away" at a later time. Therefore, in transient analyses, the solution should be carefully checked once the mesh is repaired. Some times, use of the second-order mapping scheme may also help on the accuracy. Note that the second-order mapping scheme does not affect the solution accuracy in steady-state analysis.

For steady-state solutions with moving meshes, the final solution may not be able to reached in one or two steps. The key is, starting with a coarse mesh, to refine the mesh *gradually* in steps. Similarly, the mesh should be repaired *earlier* before it becomes too distorted. Consider FSI as an example. A major difficulty arises when the required fluid element sizes are very small, as in boundary layers, and the structural deformations are rather large in an incremental step, measured on these fluid element sizes. In such cases, the fluid mesh can become very distorted or, more severely, element overlapped before converged solutions are obtained. A good strategy is, to start the solution with a rather coarse mesh and allow the large structural deformations to take place. Then, thereafter, the fineness of the fluid mesh is increased. Since the structure will in this later phase only adjust its deformations by relatively small changes in geometry, required small fluid sizes, specifically in boundary layers, can now be accommodated in the mesh adaptation to obtain the required solution accuracy.

In the FSI solutions, we also use the arbitrary-Lagrangian-Eulerian formulation for the fluid as usual, to allow reasonable mesh deformations in each solution step.

If geometries of fluid models are imported into AUI from a Nastran file, there is one restriction for the Nastran file. As known, Nastran files only contain mesh information, but no geometry information. Therefore, for such a model, SAM will do re-meshing based on the information of the element groups. This requires every boundary should be grouped as a single shell element group. For example, a model imported into AUI from a Nastran file has 4 external boundaries and 2 internal boundaries. Users should group each boundary as a different shell element group. Therefore totally 6 shell element groups should be defined in the Nastran file.

12.15.8 Adaptive Meshing Modeling Tips

What follows is a brief discussion about meshing as it relates to mesh adaptation. Focus is given to the 3-d case where interfaces are body faces, but the same principles apply to the 2-d case where interfaces are face edges.

Properly dealing with nodal coincidence is of prime importance when modeling bodies with interfaces. If the flow is supposed to pass through, nodes should not be duplicated across the interface. However, if there is no intended flow, nodes should be duplicated across the interface. This is routinely handled by the nodal coincidence parameter in the body meshing commands.

The best way to handle situations where you have some interfaces (body faces) with flow passing through and others with no flow is to use the 'SELECTED' nodal coincidence option. When meshing, this option checks if a new node already exists among the nodes on the (previously meshed) entities (typically, body faces) defined in the provided 'DOMAIN'. So, if you want flow passing only through some of the interfaces between two bodies, you should add those interfaces to the domain (when meshing).

Another important consideration is to make sure that nodal coincidence checking is not order-dependent, in other words, the order in which the bodies are meshed should be irrelevant. This means in particular that the 'COINCIDE=NO' directive should not be used. In the case mentioned in the above paragraph, all the meshing commands involving such interfaces (some with flow and some without) should make use of the same type of nodal coincidence checking so that the meshing is order independent.

Thin solid structures should be avoided when modeling fluid-structure interactions, whenever possible. It is always preferable to reduce the dimension of the structural part. A thin structure should be modeled with shell elements in three dimensions and with beam elements in two dimensions. This is to avoid self-intersections of the fluid boundaries when the mesh moves around the thin structures.

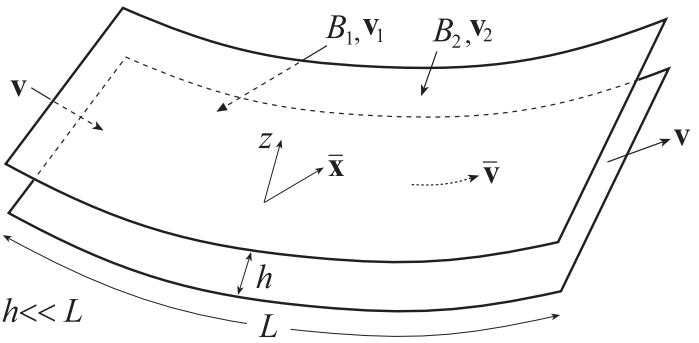
12.16 Reynolds Equation for lubrication

Reynolds Equation (RE) describes the flow of a thin lubricant film between two boundaries. It is derived from the Navier-Stokes equations and is one of the fundamental equations of the classical lubrication theory.

Currently the RE model is only available for FCBI-C elements in ADINA-CFD.

12.16.1 Reynolds Equations for smooth boundaries

Let the two boundaries, B_1 and B_2 , move in speeds $\mathbf{v}_1$ and $\mathbf{v}_2$ respectively. The distance h between the two boundaries is assumed to be far smaller than the geometrical scale in the flow direction.



In this case, the classical Reynolds equation is derived from the Navier-Stokes equations and the equation of continuity, given by

$$\frac{\partial \rho h}{\partial t} + \bar{\nabla} \bullet \left(\rho h \bar{\mathbf{v}} \right) = 0$$

where, $\bar{\mathbf{v}}$ is the average velocity

$$\bar{\mathbf{v}} = \tfrac{1}{2}\left(\mathbf{v}_1 + \mathbf{v}_2 \right) - \tfrac{h^2}{12\mu}\left(\bar{\nabla} p - \bar{\mathbf{f}}^B \right)$$

$\overline{\mathbf{f}}^{B}$ is the averaged body force per unit volume, and $\overline{\nabla}$ is the gradient operator ∇ in the plane that is perpendicular to the h-direction .

For using the smooth-boundary RE model, the following procedure must be followed:

- Generate the CFD model as usual
- Define "Boundary-Distance Condition" (BDC) (see description in conditional loads), which leads the two boundaries, B_1 and B_2, specified. This condition allows the program be able to automatically compute the boundary distance h that is one of the most important parameter in the RE model. In this condition, the open and close values (h_{open} and h_{close} respectively) are also defined. It controls the specific location where and how the RE is used. At the location where $h \le h_{close}$, RE is used. At the location where $h > h_{open}$ the model equation, usually Navier-Statokes Equation (NS), is used. In between, a hybrid model that mixes RE and NS is used. To allow RE model is fully employed, a large close value must be specified such that it is always larger than the boundary distance during the whole modeling process.
- Define RE model using "Fluid Control Model 1" (FCM1), in which set the boundary property as *smooth*.
- Apply the defined RE model to the element groups where needed.
- Note that one layer of element in the RE region is preferable (for both of computational speed and solution accuracy), although more number of layers are acceptable (for the place that requires more layers to connect outer elements).

12.16.2 Reynolds Equations for rough boundaries

With the assumption of the rough boundaries (asperity), the averaged velocity is modified to be (see reference followed)

$$\overline{\mathbf{v}} = \tfrac{1}{2}\left(\mathbf{v}_1 + \mathbf{v}_2\right) + \tfrac{1}{2}\left(\mathbf{v}_1 - \mathbf{v}_2\right) H^{-1}\phi_s - \tfrac{h^2}{12\mu}\left(\boldsymbol{\varphi}_p \bullet \overline{\nabla} p + \overline{\mathbf{f}}^B\right)$$

where, ϕ_s and $\boldsymbol{\varphi}_p$ are the shear flow factor and pressure flow factor respectively. They are expressed in the forms

$$\phi_s = \left(\tfrac{\sigma_1}{\sigma}\right)^2 \phi\left(H, \gamma_1\right) - \left(\tfrac{\sigma_2}{\sigma}\right)^2 \phi\left(H, \gamma_2\right)$$

$$\boldsymbol{\varphi}_p = \varphi\left(H, \gamma\right)\mathbf{e}_v\mathbf{e}_v + \varphi\left(H, \gamma^{-1}\right)\left(\mathbf{I} - \mathbf{e}_v\mathbf{e}_v\right)$$

where,

$\sigma_i =$ the standard deviation of roughness of $B_i \ (i = 1, 2)$

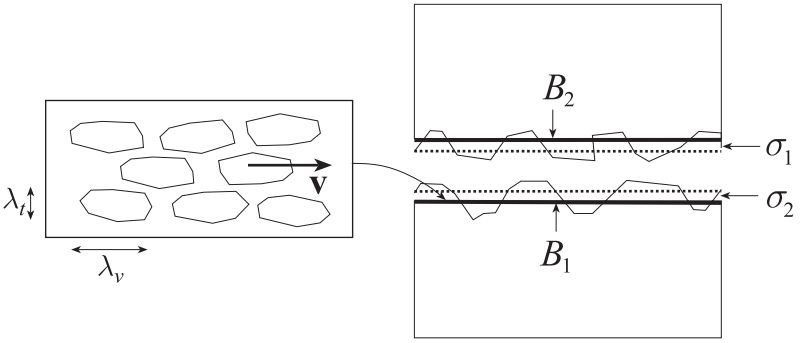
$\sigma = \sqrt{\sigma_1^2 + \sigma_2^2}$, composite roughness of the two boundaries

$H = \frac{h}{\sigma}$

$\gamma_i =$ surface pattern factor for shear flow of $B_i \ (i = 1, 2)$

$\gamma =$ surface pattern factor for pressure flow

$\mathbf{e}_v =$ uniform vector in the flow direction



The surface pattern parameter γ is defined as the ratio of correlation lengths in velocity direction (λ_v) and transverse direction (λ_t) respectively. The two functions are fitted into empirical relations of the form

$$\phi(H,\gamma_i) = \begin{cases} A_i H^{\alpha_{i1}} e^{-\alpha_{i2}H+\alpha_{i3}H^2} & for\ H \le 5 \\ A_i^* e^{-\alpha_{i4}H} & for\ H > 5 \end{cases}$$

$$\varphi(H,\gamma) = \begin{cases} 1 - Ce^{-rH} & for\ \gamma \le 1 \\ 1 + CH^{-r} & for\ \gamma > 1 \end{cases}$$

where, $A_i^* = \phi(5,\gamma_i)e^{5\alpha_{i4}}$. The set of constants $(A_i, \alpha_{i1}, \alpha_{i2}, \alpha_{i3}, \alpha_{i4})$ depends on the stress pattern factor γ_i of B_i, and the two sets of constants, (C_x, r_x) depends on the pressure pattern factor γ (in flow direction) and (C_y, r_y) depends on γ^{-1} (in transverse flow direction). For details of these parameters, see

ref. Nadir Patir and H. S. Cheng, "An Average Flow Model for Determining Effects of Three-Dimensional Roughness on Partial Hydrodynamic Lubrication", Journal of Lubrication Technology, 100 (1978) 14-17.

ref. Nadir Patir and H. S. Cheng, "Application of Average Flow Model to Lubrication Between Rough Sliding Surfaces", Journal of Lubrication Technology, 101 (1979) 220-229.

A few notes are worth mentioning for Patir-Cheng rough-boundary model

- $\sigma_i = 0$ indicates B_i is smooth
- Both $\sigma_1 = 0$ and $\sigma_2 = 0$ is equivalent to the *smooth* model
- The model result does not depend on the order of B_1 and B_2. However, the two boundaries defined in FCM1 and in the BDC must be consistent.

- A rough moving/stationary boundary accelerates/decelerates the flow.

For using the Patir-Chang rough-boundary RE model *automatically*, in addition to the setup of the smooth-boundary model, the following parameters are required in the FCM1:

- The pressure pattern factor γ
- The parameters (σ_i, γ_i) for B_i

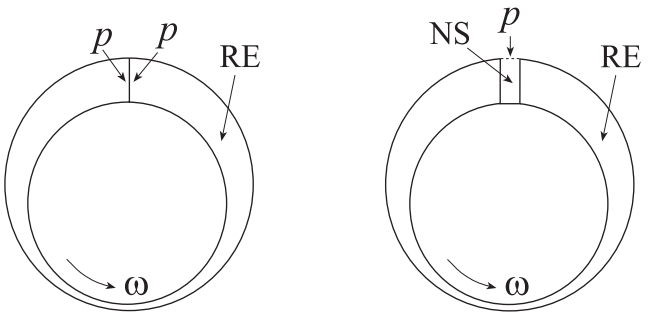
User may also directly *input* the following parameters in the FCM1:

- The pressure pattern factor γ
- (C_x, r_x) for pressure flow in flow direction
- (C_y, r_y) for pressure flow in transverse flow direction
- The standard deviation of roughness σ_i
- The constants $(A_i, \alpha_{i1}, \alpha_{i2}, \alpha_{i3}, \alpha_{i4})$ for stress flow

12.16.3 Modeling guidelines

The modeling of a RE model is quite different from the other fluid models, such as NS model. The below issues are worth mentioning.

- The application of boundary condition must be careful, since the Reynolds equation is defined in a domain of the space one dimension lower than the model space. For example, in the 2D Journal Bearing problem, the RE is an ordinary differential equation (1D problem). If a pressure condition at one side of the 2D model is required, the two modeling methods are shown in the figure below.



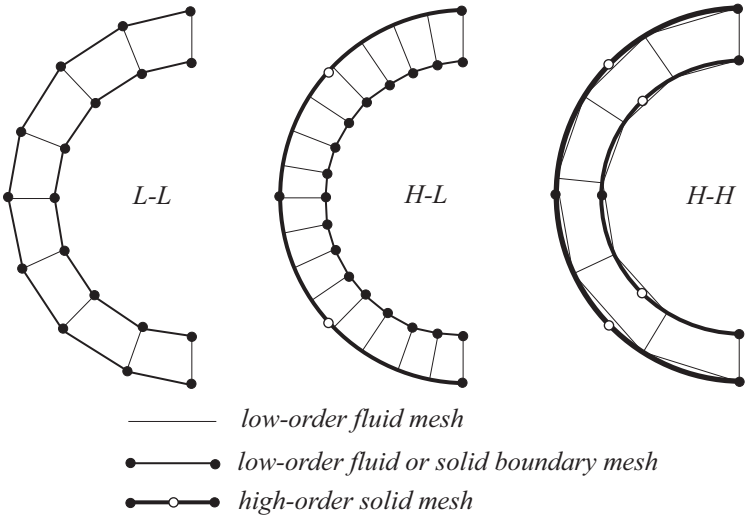
- The discretization of two curved boundaries (including FSI and Wall boundaries) should be matched well enough such that the boundary distance can be computed correctly. Otherwise, the fluid elements in the thin layer will be extremely distorted or even overlapped. In the followed figure, three usable boundary meshes are shown.

  In the left figure (L-L), low-order elements are used for both boundaries. In this case, the two meshes should be closely matched. Note that one layer mapped fluid elements are always matched.

  In the middle figure (H-L), the left boundary is represented by high-order solid elements, while the right boundary is by low-order elements. In this case, the low-order elements should be fine enough to represent the curve.

  In the right figure (H-H), both boundaries are represented by high-order solid elements, so the fluid mesh can be more freely generated.

  It is understandable that, if the two boundaries move in different speed, the boundary meshes will not matched during computation. It is then highly recommended to use high-order solid elements, if the two boundaries are solid and moved in different (transverse) speeds.



- In the RE region, one layer of elements is preferable. In case that it has to be connected properly with other elements, more layers of elements can also be acceptable.

- Recalling that the velocity in RE element group is the "averaged" velocity between the two boundaries, it is normal that the prescribed wall or solid velocity is not shown in the nodal velocity plot.

12.17 Universal Barotropic Cavitation model

The Universal Barotropic Cavitation (UBC) model, resembling the model proposed by Cooper, is a simplified description of the vaporization and condensation two-phase flow.

ref. Cooper P., 1967, "Analysis of single and two-phase flows in turbopump inducers", J. of Eng. for Power, pp 577-588.

Ideally, the mixture is supposed to be purely liquid when the pressure is over the vapor pressure and purely vapor at pressure lower than the vapor

pressure. The two extreme phases are joined continuously by a single varying pressure-dependent density

$$\rho = \rho_v + f\left(x\right)\left(\rho_l - \rho_v\right)$$

where, ρ_l and ρ_v are densities of liquid and vapor respectively, $f\left(x\right)$ is the function of x, and x is a nondimensional representation of the pressure

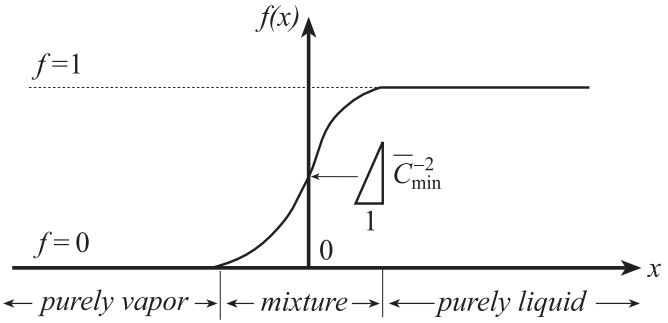
$$x = \frac{p - p_v}{p_d + p_v}$$

In which, p_v is the vapor pressure of the model, p_d is the datum pressure of the model such that $p_d + p_d$ represents the absolute vapor pressure. The function is obtained in an analytical fashion, with only one unknown parameter to be input: the minimum sound speed $C_{\min}$. The sound speed in the cavitation region (a mixture of liquid and vapor) could be dramatically lower than the sound speeds in either liquid or vapor regions. In fact, its true magnitudes in various vapor-liquid mixtures are still unclear to scientists and engineers. In numerical models, its value can be treated as a calibration parameter.

The required minimum mixture sound speed $C_{\min}$ has been made dimensionless in the input

$$\overline{C}_{\min} = C_{\min}\sqrt{\frac{\rho_l - \rho_v}{p_d + p_v}}$$

Note that $\overline{C}_{\min}^{-2}$ represents the maximum slope of the function $f\left(x\right)$ in the mixture region.



The basic procedure for using the UBC is described below.

- Generate the CFD model as usual, in which the fluid is assumed to be "liquid"
- Define the additional material of "vapor" (M2)
- Define UBC using "Fluid Control Model 2" (FCM2), in which the fluid properties of the mixture are specified, including vapor pressure p_v pressure datum p_d and the nondimensional minimum sound speed $\overline{C}_{\min}$.
- Apply both M2 and FCM2 to the element groups where the cavitation phenomenon is expected.

The followings are a few notes for using UBC

- The selection of the parameter $\overline{C}_{\min}$ varies for different flow models. A value between 0.5 and 20 can be a starting point. The numerical solution is normally more stable with larger value, but is sharper with smaller value, until diverges when it is below the smallest critical value. Usually, a value slightly above the critical value should be a good selection.
- Although the UBC model can dramatically improve the solution in cavitation region, it cannot guarantee the absolute pressure obtained is always positive if $\overline{C}_{\min}$ is too small.

- Since the UBC only changes the density (e.g., the state equation), it can be used for incompressible flows (as defined in the primary flow model) using Navier-Stokes equation, Reynolds equation and Darcy equation.

- Currently the UBC model is only available for FCBI-C elements in ADINA-CFD.

Chapter 13 Other topics

13.1 Model preparation and testing

Before any model is established and analysis performed, the following issues should first be resolved:

- What is the suitable fluid model (incompressible, slightly compressible, low speed compressible, high speed compressible or porous medium)?
- Should a steady-state analysis or a transient analysis be performed?
- What time/load step size should be used?
- Which part of the geometric domain should be modeled?
- What boundary conditions are most adequate?
- What element and mesh should be used?
- Which material model describes the fluid behavior most adequately?
- What is the procedure to use to reach a successful solution of a complicated problem?

To answer the above questions requires an understanding of the physical problem and familiarity with ADINA-F, including the available capabilities, assumptions and limitations.

13.1.1 Choosing the fluid model

Most engineering applications of fluid flows are incompressible flows. If a domain with open boundaries or an enclosure with rigid boundaries is considered, the incompressible flow model is generally applicable. Examples of fluids are oil, water, air (when the Mach number is less than about 0.3), etc.

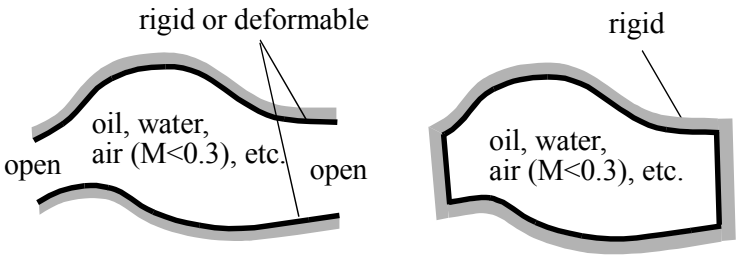


Figure 13.1 Typical problems that incompressible flow model can be applied

If the boundary of a domain is fully closed, the flow is called confined or flows in an enclosure. If the boundary of confined flow is deformable, as in a fluid-structure interaction problem or in problems where boundary displacements are prescribed, the fluid cannot be treated as fully incompressible.

Oil and water generally show negligible dependency of density on temperature, so it is adequate to model their flows using the slightly compressible flow option. When the boundary moves very slowly, the air can be treated as a slightly compressible flow as well.

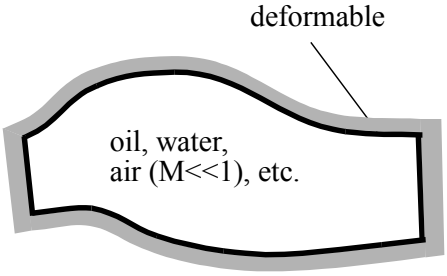


Figure 13.2 Typical problems that slightly compressible flow model can be applied

In many applications, the compressibility of air must be taken into account. The problem must distinguish between low-speed and high-speed fluid flows purely because of numerical limitations. Typical examples of

low-speed compressible flows are those pertaining to enclosures whose boundaries are deformable.

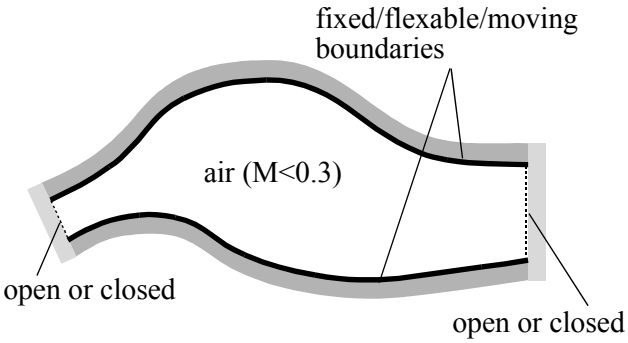


Figure 13.3 Typical problems that low-speed compressible flow model can be applied

Whenever the Mach number is of order 1, the flow must be treated as a high-speed compressible flow. Most flows of this type are open to the environment.

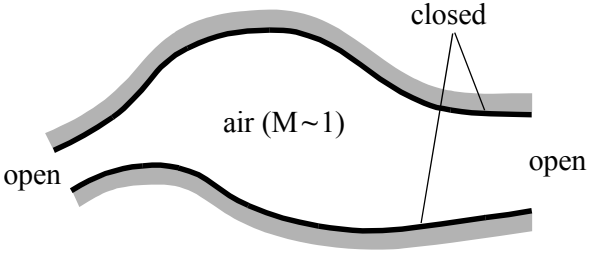


Figure 13.4 Typical problems that high-speed compressible flow model can be applied

The porous medium flow model has wide applications. In addition to flow through porous solids such as soils, it can also be used to model many other problems such as those involving air filters with small or large holes (such as cigarettes). It is worth noting that the model can be incorporated

with incompressible, slightly compressible and low-speed compressible flow models through element group options.

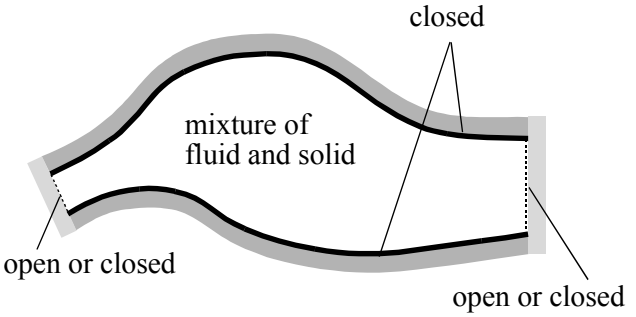


Figure 13.5 Typical problems that porous media flow model can be applied

When many small holes in a model need to be modeled simultaneously, a porous medium model is the best choice. As shown in the following figure, we can use a porous medium element group to simulate the pipes instead of modeling the small pipes directly (which may not give good results depending on the capacity of the computer used).

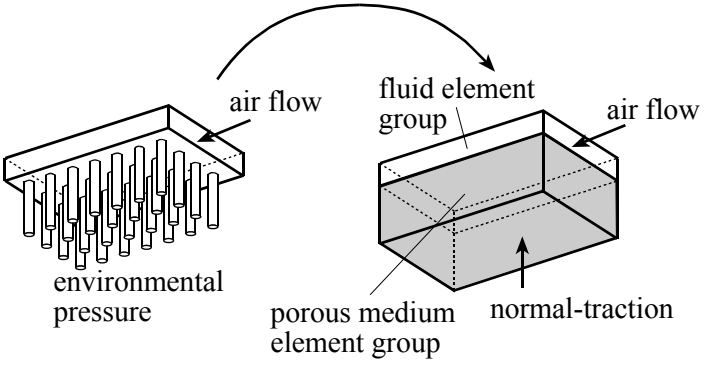


Figure 13.6 Example of using porous media flow model

13.1.2 Choosing the analysis type

If a steady state physically exists, either a steady-state analysis or a transient analysis can be performed to reach the steady-state solution. To model such a problem, all materials, boundary conditions, etc., must be time-independent at the last time step at which the steady solution is expected. Recall that the definition of a steady-state analysis in ADINA-F is to drop the dynamic terms $\partial/\partial t$ in the governing equations. We have the following ways to reach a steady-state solution.

- Perform one time step using a steady-state analysis and all materials and boundary conditions are time-independent. This method can be applied to some simple flow problems. The computational effort can be demonstrated in the following chart:

$$\begin{cases} (1b)\ \textit{for equilibrium iteration}\ 1,\ 2,\ \ldots \\ \quad \textit{solve the linearized steady-state equation} \end{cases}$$

- Use a proper CFL number associated with the method (1). Note that the CFL option is, in terms of numerical stability, equivalent to the time marching method in a transient analysis that reaches a steady-state solution. The advantage of using the CFL option is to let the program decide the proper auxiliary "time step" and the converged solution is the steady-state solution. This method is recommended in almost all fluid flow problems. In particular, method (1) can be obtained if the CFL number is very large. The computational effort can be demonstrated in the following chart:

$$\begin{cases} (2b)\ \textit{for equilibrium iteration}\ 1,\ 2,\ \ldots \\ \quad \textit{solve the linearized unsteady-like equation} \end{cases}$$

The advantage of iteration (2b) is that the matrix has a better conditioning. With a properly chosen CFL number, the approach can give a faster convergence as well. However, the CFL number cannot be too small. The value 1 should be regarded as the lower limit for implicit methods.

- Perform a few time steps using the steady-state analysis option and time-dependent materials or boundary conditions. The time step here does not correspond to the real time; hence they are rather called load steps. The solutions obtained at each time step are in steady state corresponding to the conditions at that time. The solutions obtained from the previous time step serve as initial conditions for the next time step. This method can be applied to some flow problems with flow transition stages. Typical examples are natural convection, flow in curved channels, or any flows where fluid circulation zones may occur. The solutions obtained at each time step are sometimes required to represent solution sequences under different load conditions (represented by specific Reynolds numbers, Rayleigh numbers, etc.). The computational effort can be demonstrated in the following chart:

$$\begin{cases} (3a)\ \textit{for time step}\ 1,\ 2,\ ...,\ N_t \\ \quad \begin{cases} (3b)\ \textit{for equilibrium iteration}\ 1,\ 2,\ ... \\ \quad \textit{solve the linearized steady-state equation} \end{cases} \end{cases}$$

  Usually, iteration (3b) has a better initial condition than (1b). Although more time steps are performed, the overall CPU time may be less. Above all, (3b) may converge in problems for which (1b) diverges.

- Use a proper CFL number associated with the method (3). The computational effort can be demonstrated in the following chart:

$$\begin{cases} (4a)\ \textit{for time step}\ 1,\ 2,\ ...,\ N_t \\ \quad \begin{cases} (4b)\ \textit{for equilibrium iteration}\ 1,\ 2,\ ... \\ \quad \textit{solve the linearized unsteady-like equation} \end{cases} \end{cases}$$

  Iteration (4b) has a better matrix conditioning than (3b) if a proper CFL number is used. However, as explained for (2b), when the CFL number is too small, convergence may become slow. Nevertheless, this is the most frequently used method for steady-state analyses.

- Use the time marching method, in which a transient analysis is performed, to reach the steady-state solution. The steady-state solution may be obtained during the time step marching or, in addition, a restart

run must be performed. Since only the final steady solution is of interest, the equilibrium iterations can be omitted. The computational effort can be demonstrated in the following chart:

$$\begin{cases} (5a)\ \textit{for time step}\ 1,\ 2,\ ...,\ N_t \\ \quad \textit{solve the linearized unsteady equation} \end{cases}$$

The equilibrium iteration can be omitted by choosing either a large iteration convergence tolerance or 1 as the maximum iteration number. A steady-state tolerance can be specified as well to ensure that a steady-state solution is reached. However, in the event that the steady-state solution has not been reached, the final solution is always saved and stored in the porthole file and in the restart file, so that the solution can be visualized (for example, to check if indeed the steady-state solution has been obtained) and, if necessary, a restart-run can be performed.

When problems become unsteady in nature, transient analyses must be performed. The initial conditions in this case must be actual physical conditions. Physically meaningful time steps must also be chosen (see Section 13.1.3). When time steps are large, convergence in equilibrium iterations must be enforced for an accurate transient solution. In this case, the computational effort can be demonstrated in the following chart:

$$\begin{array}{l} (6a)\ \textit{for time step}\ 1,\ 2,\ ...,\ N_t \\ \quad \begin{cases} (6b)\ \textit{for equilibrium iteration}\ 1,\ 2,\ ... \\ \quad \textit{solve the linearized unsteady equation} \end{cases} \end{array}$$

Remember that the truncation-error in time is always the first order (for $\alpha > 1/2$). Hence, the accuracy in time frame is proportional to the time step size. It is better to always iterate for equilibrium in implicit transient analyses unless the CFL number is small (for example, in the order of 1). If the time step size is very small, the user may dispense with the equilibrium iterations but then the results must be carefully checked. The procedure can be achieved by choosing either a large equilibrium iteration tolerance or 1 as the maximum iteration number. In this case, the computational effort is the same as given in the method (5).

13.1.3 Choosing the time/load step size

In steady-state analyses, the time step size has no physical meaning. The time functions determine the conditions at each time steps.

However, in transient analyses, time steps are physical parameters. Therefore, the time step sizes must be reasonable to visualize the solution as well as to be able to accomplish the solution. Implicit methods have no limitation of time step sizes in order to be numerically stable but convergence in the equilibrium iterations must be reached. The choice of the time step can solely be based on the physical conditions. The time steps should be determined so that the solutions can represent the physical phenomena of interest.

Many transient problems are fundamentally periodic in nature. This property can be predicted based on the following: experience with the problem, experimental data that is available, periodic boundary conditions that are applied, a previously computed solution to a similar problem, analytical results from a simplified problem, etc. A reasonable choice of the smallest time step size for such problems is about 1/20th to 1/100th of the smallest period of interest, depending on the accuracy that is sought and the CPU time available.

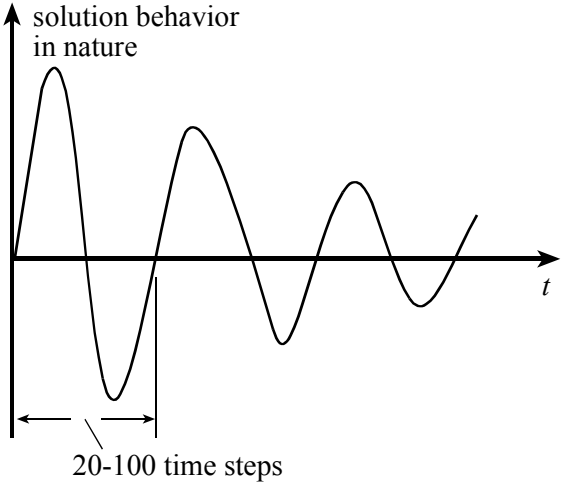


Figure 13.7 Time step sizes in periodic solutions

Some transient problems are fundamentally determined by the waves traveling in the domain of interest. For slightly compressible flows, the wave speed can be predicted by $w = \sqrt{\kappa/\rho}$ and for compressible flows, the wave speed can be predicted by the sound speed $w = c = \sqrt{\gamma p/\rho}$. A certain number of time steps is required in order to capture the wave motion. Typically, 10 time steps are the minimum needed to allow the wave to travel a distance L that characterizes the computational domain. Therefore, a time step size of $L/(10w)$ should be regarded as the largest that is allowed. On the other hand, too many time steps are unnecessary as well. A CFL number 1 should be regarded as the minimum time step size that can be used. In other words, the time step should not be smaller than $\Delta x/w$.

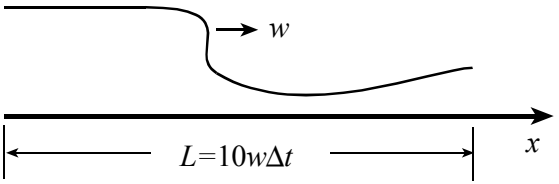


Figure 13.8 Time step sizes in wave propagation problems

There are many problems that are fundamentally diffusive in nature. Small Reynolds numbers and small Peclet numbers characterize these problems. The nature of these problems is their similarity described by the dimensionless variable $\eta \sim x/\sqrt{2Kt}$, where K is the diffusion coefficient that represents μ/ρ in the momentum equations, $k/\rho C_v$ in the heat transfer equation, etc. In a one dimensional diffusion problem, for example, the solution can be regarded as steady when $\eta > \eta_{\max} \approx 3$. We can perform N_t time steps of equal size $\Delta\eta$ (i.e., $\Delta\eta = \eta_{\max}/N_t$) to cover the solution resolution. The maximum time is therefore evaluated as $t_{\max} = L^2/2K\Delta\eta^2$ and, accordingly, the time steps can be determined as $t_n = t_{\max}/(N_t - n + 1)^2$. We can see that the time step sizes ($\Delta t_n = t_{n+1} - t_n$) vary quadratically from the smallest one, where the solution changes the most to the largest one, where the solution is considered in steady state.

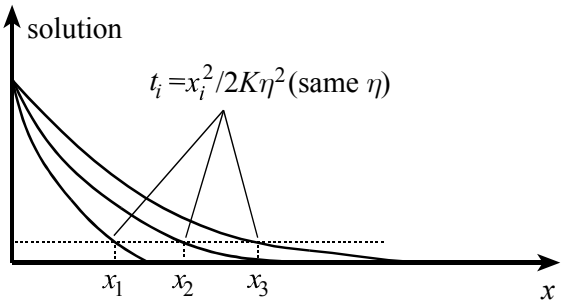


Figure 13.9 Time steps in diffusion problems

When problems become more complicated in nature, neither of the above factors is dominant and a natural choice is the smallest time step of the possibilities if the whole history of the solution is required.

It is worthy to mention here again, whenever the time step is small (say the CFL number is about order 1), the equilibrium iteration may be omitted. Particularly when a large number of time steps are performed, this approximation can save tremendous CPU time (typically 50% to 90%).

13.1.4 Choosing the computational domain

Obviously, we cannot include any details or sizes in a model. The model must only contain the relevant data and phenomena of physics that are suitable for the capacity of the computer. Some guidelines are listed below.

- The domain must be "cut" somewhere from the "rest of the world". Reasonable boundary conditions should be available to represent the flow conditions on the boundary. For example, when a circulation zone is expected behind a building, a long region should be used there if a zero normal-traction is applied. If the circulation zone cannot be included completely, frequently a transient analysis needs to be performed.
- When objects are small enough to be negligible compared to the major parts of the model, they should be omitted or modeled in different ways.

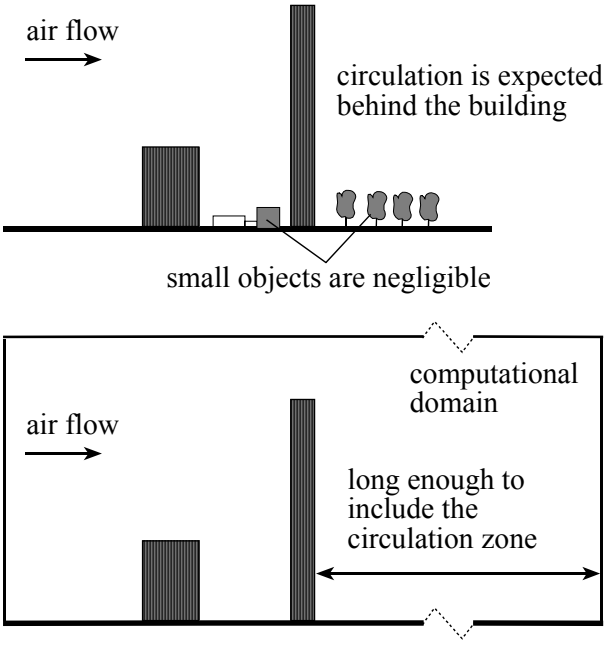


Figure 13.10 Choosing the computational domain

- Avoid simulation of one-dimensional flows in two- or three-dimensional models. Some one-dimensional parts can be cut from the computational domain or modeled with a porous medium element group. For example, in the following figure, the pressure drop in the one-dimensional part can be evaluated by an analytical solution or by the solution of modeling a portion of the one-dimensional part with a given fluid velocity. The pressure based on the one-dimensional analysis is then applied to the outlet of the separated three-dimensional model. Sometimes the one-dimensional part can be modeled using a short porous medium element group as well.

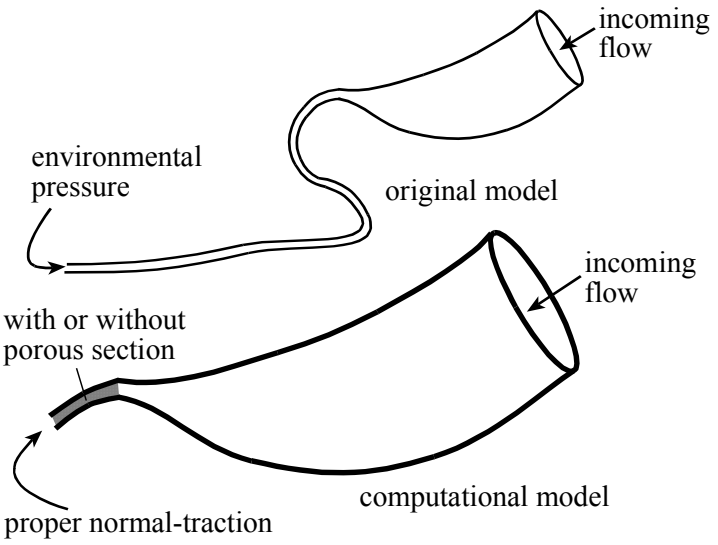


Figure 13.11 Avoid lower dimensional flows in the model

- A similar situation happens when a porous medium is too thin compared with the model dimension. One can actually model it using a thicker porous medium with a modified permeability. This procedure is equivalent to using different units in the porous media domain.

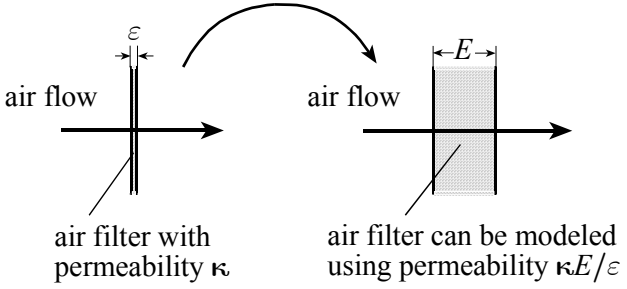


Figure 13.12 Using equivalent physical models

- Special attention should be given to modeling external flows. The physical boundary of such a problem is infinitely far away from the region where the action takes place. The finite element model cannot have an infinitely large domain; hence a truncated domain must be used. However, the truncated (computational) domain should be large enough to represent the infinity and, accordingly, represent the fluid flow in the region of interest accurately.

13.1.5 Choosing proper boundary conditions

The use of appropriate boundary conditions is probably the most important task in computational fluid dynamics. In addition to the general considerations given for each flow model as described in the corresponding chapters, here are some other guidelines on using ADINA-F.

- When a pressure condition is to be specified at an open boundary, apply a distributed normal-traction, rather than a prescribed pressure. A normal traction is equivalent to external force acting on the boundary, so the velocity and pressure are solved by the governing equations. The pressure obtained from the solution will be slightly different from the value of the normal-traction that is applied, which is reasonable along open boundaries. On the other hand, if a prescribed pressure is enforced, the discretized continuity equations on that boundary are removed from the governing system. Then nonphysical phenomena could be generated near the boundary.
- In an incompressible flow solution, one open boundary can be left free (no traction/pressure is specified), which corresponds to a zero normal-traction. However, if multiple open boundaries are subjected to different pressure conditions, the proper normal tractions must be applied to each of these open boundaries (see the figure below).

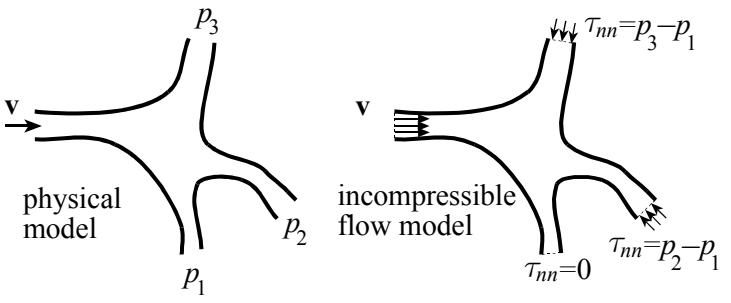


Figure 13.13 Multiple normal-traction conditions in incompressible flow models

- In compressible flows (in both low- and high-speed compressible flow models), no zero normal-traction should be applied. The absolute pressure value must be specified. Note that the same conditions can also be applied to incompressible flows for the case considered in Fig. 13.13. In high-speed compressible flows, the pressure conditions are usually defined and no additional normal-tractions should be applied.

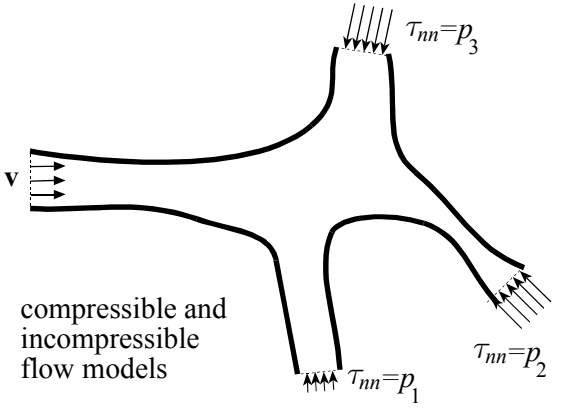


Figure 13.14 Multiple normal-traction conditions in compressible and incompressible flow models

- In confined incompressible flows, such as those in natural convection problems, a fixed pressure must be applied to each of the singly connected domains, see the following figure.

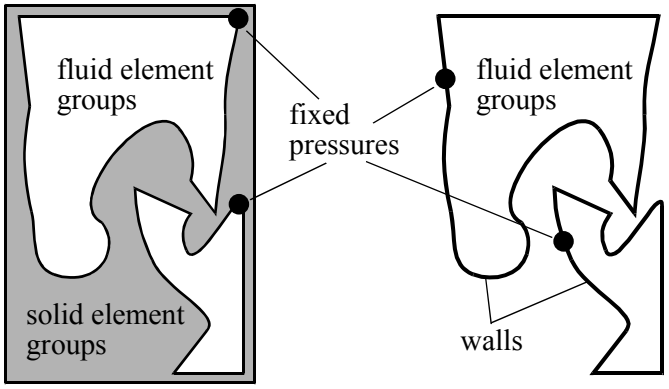


Figure 13.15 Fixed pressure condition must be applied to confined incompressible flows

- Use special boundary conditions as much as possible in high-speed compressible flow analyses. Remember that the prescribed conditions are always applied to the conservative variables, rather than to the primitive variables. For example, in internal flows, subsonic-inlet, subsonic-outlet and supersonic-outlet are frequently used. In external flows, external conditions are frequently used. If a prescribed velocity in a certain direction, say **a**, is absolutely necessary, a few choices are available. They are listed here.

    1) If the normal direction of the boundary **n** corresponds to **a**, apply the subsonic-inlet condition with the "*slipc*" parameter zero. The thermal condition can also be specified as either a prescribed temperature or a zero heat flux.
    2) Apply a "wall" condition of type "tangential" with the tangential direction specified as **a**.
    3) Assuming a zero first derivative of density, an external condition of type (θ, ρ, v), with specified velocity, zero heat flux or prescribed temperature, and zero derivative of

density, can be applied (using *slipc*=0 to enforce the flow direction).

4) Assuming a zero first derivative of pressure, a supersonic condition of type (p,θ,v), with specified velocity, zero heat flux or prescribed temperature, and zero derivative of pressure, can also be applied (using *slipc*=0 as well).
5) When the velocity is the result of a boundary condition from a rotating disk, for example, a wall condition of type "*rotational*" must be applied.

- Do not apply a moving boundary condition if the domain deformation is not of interest. Instead, a proper boundary condition applied on a fixed boundary may be more adequate. For example, a circular laminar jet produces a free surface that represents the front of the jet flow. If a free surface is specified (that is, a Lagrangian formulation is applied to the free surface while an ALE formulation is applied to the whole domain), the solution is difficult to reach. In fact, the steady-state solution cannot be obtained in this model. Instead, a reasonable boundary condition set applied to an axisymmetric plane can be used to well approximate the problem, see the following figure.

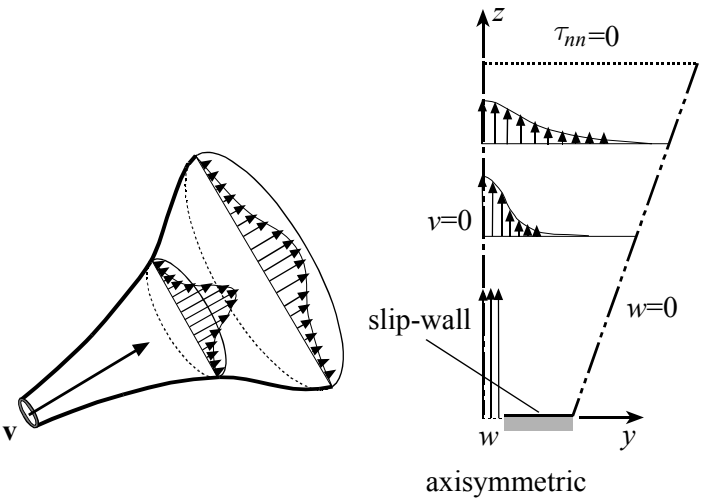


Figure 13.16 Solution of a circular laminar jet

13.1.6 Choosing the proper element

The 2D 3-node triangular and the 3D 4-node tetrahedral elements are suitable for high Reynolds number and high Peclet number flows, although they can also be used for low Reynolds number and low Peclet number flows as well. For high-speed compressible flows, only these two elements are available. The 2D 9-node quadrilateral and 3D 27-node brick elements are very good elements for highly viscous flows, characterized by low Reynolds number and low Peclet number flows. The 6-node triangular element is usually used in conjunction with the 9-node quadrilateral element as its triangular counterpart. Only the 9-node quadrilateral element and the 6-node triangular element are compatible and therefore can be used together.

On the other hand, the FCBI elements can be successfully applied to all incompressible, slightly compressible and low-speed compressible flows. These elements are very accurate provided the element quality (aligned to the flow direction, smoothly varying in size, etc.) is reasonably good.

To obtain an accurate solution, the mesh size should be sufficiently fine, especially in regions where there are high solution variable gradients. Such regions usually occur near solid boundaries, circulation zones and shocks in high-speed compressible flows. However, the number of elements is also limited by the capacity of the computer in terms of storage and speed. For a fixed number of elements that allows the analysis to be completed with reasonable resources, in general, the elements should be distributed such that $\Delta x \|\nabla f\| \approx \text{const.}$, where Δx is the element size and f is a target solution variable. This variable could be the pressure in high-speed compressible flows where shocks are fundamentally important phenomenon to be modeled; it also could be the velocity for flows in boundary layers; etc. Larger element sizes should be used in regions where a smooth varying solution is expected. On the other hand, wherever a sharply changing solution is expected, the element sizes should be fine enough to capture the solution gradients. Therefore, varying mesh sizes are generally used in the discretization of the computational domain. However, the transition of the element sizes from fine to coarse should be as smooth as possible. For an optimal mesh, the gradient of the element sizes should be approximately proportional to the magnitude of the second derivatives of the solution variable.

13.1.7 Choosing the material model

The selected material model should pertain to the fluid that is modeled. The material constitutive relations may be constant, time-dependent, temperature-dependent, etc. A proper consistent unit system must be used to avoid round-off errors. Here the feature of automatic nondimensionalization in the fluid model of ADINA-F can be very useful. In case gravity effects must be included, consider the option to remove hydrostatic pressure.

13.1.8 Choosing the solver

There are basically two types of solvers that can be selected: direct solver (sparse or SKYLINE) and iterative solvers (RPBCG, RPGMRES or Multi-grid).

The Gauss elimination method (SKYLINE) should not be used for any problem except for research studies.

If the matrix conditioning is always good (in the solution time range), the iterative solvers are the best choices. Flows of low Reynolds/Peclet/Rayleigh numbers, FSI problems with small displacements, etc. could be such problems. Even for some sensitive problems the nondimensionalization can be used to improve the matrix conditioning (see Section 13.3). Note that improper units can cause failure in convergence using the iterative solvers even for physically stable problems. The ratios of the factorized diagonal elements are also a good indicator of a good/bad matrix conditioning.

Usually a few hundred iterations performed with an iterative solver is reasonable if the number of equations is less than 100,000. When the number of equations becomes larger (say around 500,000 equations), thousands of iterations may be required.

In all other cases, use the sparse solver. The sparse solver is a direct solver and hence will always give the solution of the algebraic equations. Of course, these equations may be ill conditioned and the sparse solver will give messages to that effect, and indeed indicate physical instabilities in the problem. When the conditioning of the matrix is poor, the solution from direct solvers may also be misrepresented because of the round-off errors. Use of proper units will always help in this situation.

A few numerical experiments can be performed as well to determine which solver is best for your problem.

13.1.9 Model testing

Special attention must be given to ensuring that a model with complexity has been generated correctly and with the appropriate physical parameters. There are many possibilities in which a model could have been constructed incorrectly. In order to obtain an accurate solution, testing is essential. It usually turns out that a good approach in that regard saves time as well. Here are steps that help progress toward a final model.

(1) Analyze the model with your experience, available experimental data and known similar solutions, etc. Understand the nature of the problem to be modeled. Is it a two-dimensional, axisymmetric, or three-dimensional flow? Is it an incompressible, slightly compressible, low-speed or high-speed compressible flow? Or is it a porous media flow? What is the best way to model the problem in terms of accuracy, numerical stability, computer storage and CPU available? Which factors could be fundamentally difficult in the model? Can any part of the model be simplified?

(2) If possible, test a simplified model first. For example, test a two-dimensional model before creating the three-dimensional model. In this model, follow the steps described below as well.

(3) Test the model using a coarse mesh and material data and loads that permit a quick solution (e.g., large viscosities, small velocities, small normal-tractions, etc.). The purpose of this test is to logically check the input data. In this model, the *types* of materials and the *types* of boundary conditions should be the same as those that will be used eventually. Since the model is very easy to understand and to solve in terms of physics as well as in terms of numerical computations, the solutions will be easily obtained. In case of errors in the input, they can be quickly identified and corrected as well. The solution is not expected to represent the real solution, but must be physically reasonable.

If any difficulty occurs, make sure the problem set is indeed easily solved. You may even exaggerate the case by imposing a condition that will produce a solution extremely close to the initial condition. You may also iterate only once (by specifying a large convergence

tolerance) to visualize the immediate behavior of the "solution". With the help of the information printed in the *.out and *.log files and of the visualization of the results, try to find out where the difficulty lies.

- If an incorrect immediate "solution" is found near a boundary, a boundary condition could have been applied incorrectly near that location.
- Velocity vectors that show fluid flowing into an interior location usually indicates a false connection of the elements in that region as generated in the incorrect use of the AUI. This error can also be found if the mass is obviously not conserved.
- Fluid flowing into the opposite direction than is physically expected could possibly indicate that a wrong direction in the boundary condition has been used. Those conditions could be a normal-traction (remember that the normal-traction depends on the orientation of the boundary geometry!), a negative prescribed velocity, etc. In addition, a required wall boundary condition may not be specified too.
- Check if there is any error or warning message. If there is more than one message, check the first one first. An extremely large ratio of factorized diagonals (say 10^{20}) in the first equilibrium iteration in the first time step is an indicator that the model is ill posed, or mathematically unstable. Perhaps the "to be" fixed pressure is not specified in confined flows. Zero viscosity or heat conductivity might have been used in the material data. In compressible flow models, a zero pressure or temperature condition is probably used, either explicitly specified or assigned as the default in the program.
- If a zero unfactorized diagonal corresponds to a velocity or temperature degree of freedom, the material data could be incorrect. For example, a zero viscosity or zero heat conductivity might have been specified.

1) Change some key parameters one by one to see if the solution responds reasonably. The solution behavior can also be revealed from this testing.
2) Further test the model using the real material data and loads. If difficulties occur in convergence, find out which parameter is most sensitive. Usually this can be a large prescribed velocity, a small viscosity, a large heat expansion coefficient, a large heat

flux load or prescribed temperature and so on. The difficulty is usually related to a high Reynolds number, high Peclet number, high Rayleigh number, etc. The key is not to change the model that has been tested in step 3 except for the *values* corresponding to the types of materials and boundary conditions. This excludes the possibility of creating other errors. Therefore any difficulty that occurs in this stage must not result from the input errors that could have happened and corrected in step 3. Use any of the methods, or a combination of them, introduced in Section 13.3 to achieve a converged solution. Since the mesh is coarse, the solution can also be obtained quickly. The model here should be tested thoroughly, exposing difficulties and overcoming them accordingly. At the completion of this stage, the tested model should be numerically stable and physically reasonable. We emphasize the numerical stability here, because the model in the final run (next step) could be slightly more difficult. In order to reach a stable model in the final stage, some slightly more difficult conditions can be introduced at the current stage. It is worth spending more time in this stage, tuning the model to make sure it is in good condition. In this test, knowledge in choosing time step sizes or increments of loads, reasonable CFL numbers, properly nondimensional parameters, and so on is gained.

3) Only refine the mesh in the last stage of analysis and expect a solution with high resolution. In general, a finer mesh will make the model numerically more stable and therefore easier to converge. However, in certain cases, the larger number of equations may cause large round-off errors as well. If the problem is sensitive, difficulties may occur again. If this is the case, adjust the control parameters and tested again in step 5. When iterative solvers are used, the maximum number of iterations in these solvers may need to be increased if the number of equations is very large (typically larger than 250,000). The finer mesh solution may also resolve more complicated physical phenomena, such as re-circulations, shocks etc., that could not be simulated using coarse mesh in step 5. In this case, smaller increments of loads, smaller time steps, a smaller CFL number may need to be used.

13.2 Moving mesh control

In models that have moving meshes, nodal displacements are either prescribed by means of physically moving boundary conditions or arbitrarily moved in preserving the best mesh quality. The automatic procedure for moving the mesh implemented in ADINA-F assigns displacements to the nodes that can be arbitrarily moved. However, in certain cases, the automatic procedure may not be able to control the mesh quality well. An example is shown in Fig.13.17, where a ball is moving. A typical element shown has been distorted due to the ball movement even though the computational domain is simple. The reason is that there are no moving conditions on the outer boundary of the domain, i.e., point 2 is fixed and the line 1-2 becomes largely skewed.

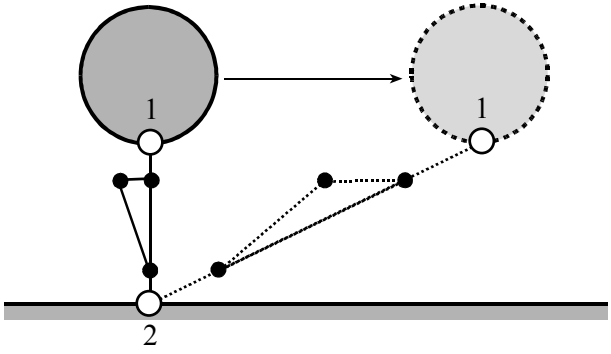


Figure 13.17 Distorted element in a moving boundary problem

If we force point 2 to follow point 1, then line 1-2 will not be skewed and the element will keep its quality. The revised situation is shown in Fig.13.18. In this case, point 1 is the leader and point 2 is the follower.

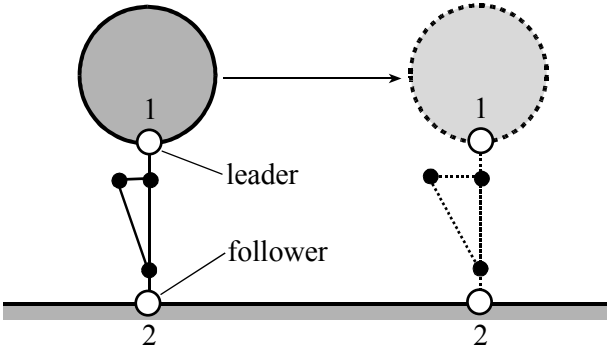


Figure 13.18 Use of leader-follower option

It is clear that the specific geometry plays a key role in the ALE formulation. In the AUI, the geometry is entered using points, lines, surfaces and volumes; the mesh is generated based on the geometry.

It is also clear that there is no guarantee that the mesh will not overlap during arbitrary movements of the boundaries. However, as demonstrated earlier, if the domain is simple and adequate, the mesh will not overlap. A good geometric domain is akin to a convex set. Recall that a convex set is a domain in which a line joining any two interior points stays in the domain. The simplest good convex domains are a "brick" and a "tetrahedron" in three dimensions and a "quadrilateral" and a "triangle" in two dimensions. For more complex domains, a good mesh for the ALE formulation can be achieved only if the domain is properly divided into simple and adequate sub-domains.

Note that a mesh overlap is likely to occur in those areas in which the deformations correspond to "compression" of the mesh. Some other helpful tips are:

- Try to create surfaces/volumes that are as close as possible to convex sub-domains.
- If a certain region cannot be converted into adequate sub-domains, you may define it as moving rigid body (that is, all the displacements of the fluid nodes in this region are the same). You can achieve this by using the leader-follower command to constrain all of the corner points of the region.

- For those regions within the domain that have regular shapes, try to create brick-like sub-domains. For the remaining region, keep them rigid. For example, if the fluid mesh near a deformable structure is likely to overlap, you may set the surrounding surfaces/volumes to move along with the structure, to minimize the deformations of the fluid mesh near the structure.
- Consider all possible mesh movements that can occur during the solution process while you design the sub-domain topology. In particular, consider critical mesh conditions in which you must control the mesh quality so that the solution is represented accurately under these conditions.
- In certain cases, in order to prevent overlap, one can use the contact surfaces (defined in the solid model). The contact surfaces may be embedded there solely for the purpose of preventing the fluid mesh from overlapping and will not affect the solution obtained. In the example illustrated below, we have added two contact surfaces that limit the region in which the deformable structure can move. The lower contact surface has a short distance from the real wall that is modeled in fluid model. This distance is small enough such that the position of the structure is always above the wall and large enough such that the fluid meshes will always be valid. If the contact surfaces are placed in proper locations, they will not affect the final solution. In transient analyses, the contact may physically happen. In this case, the upper contact surface may be a real structure. The lower contact surface would represent the physical wall.

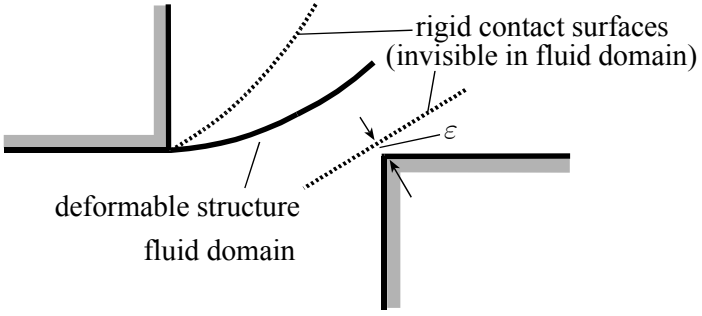


Figure 13.19 Using contact surfaces in FSI analyses

Note that the above discussion may not need to be referred to if the problem results in small displacements, say less than one element size. The automatic procedure of moving meshes using the Laplacian procedure is usually successful in the solution of such problems.

13.3 Strategies toward obtaining converged solutions

In order to obtain converged solutions, we must know what factors could make the solution divergent.

Recall that our task is to obtain the solution at time $t + \Delta t$ of the nonlinear finite element equations represented by $\mathbf{F}(\mathbf{X}) = \mathbf{0}$. We take the Newton-Raphson method (see Section 11.1) as an example here.

The convergence of the Newton-Raphson iteration strongly depends on the initial guess $\mathbf{X}^0$ that is equal to the solution calculated in the last solution step ${}^t\mathbf{X}$ or the given initial conditions. A necessary condition for the iteration to converge can be stated simply as follows:

Theorem: If the matrix $\mathbf{A}$ is not singular, there exists a set $\mathbf{D} = \left\{\mathbf{X}; \left\|\mathbf{X} - {}^{t+\Delta t}\mathbf{X}\right\| \le r,\ r > 0\right\}$ and for any $\mathbf{X}^0 \in \mathbf{D}$, the Newton-Raphson scheme converges.

This theorem indicates that, once the initial guess is close enough to the solution, the Newton-Raphson iteration will converge. On the contrary, therefore, if the final solution is very different from the initial condition, the iteration procedure may diverge.

Certain solvers are used to solve the linearized equation. The sparse solver is a direct solver that is based on the Gauss elimination method. The iterative solvers are also linked to the Gauss elimination method through their preconditioned matrices generated in the incomplete Cholesky factorizations. Without mentioning other round-off errors that may occur in the solution, the Gauss elimination method introduces round-off errors that can be significant when the problem is ill-conditioned, expressed by the condition number $\left\|\mathbf{A}\right\|\left\|\mathbf{A}^{-1}\right\|$, where we have not used the row/column interchanging technique (for the purpose of efficiency in computing), the round-off errors may be out of control and lead to a wrong solution or

destroy the convergence of the whole iteration. More specifically, what we have solved is actually the solution of the perturbed equation

$$\Delta\hat{\mathbf{X}}^k = (\mathbf{A}+\mathbf{a})^{-1}(\mathbf{B}+\mathbf{b}) \tag{1.16}$$

where $\mathbf{a}$ is a matrix containing the round-off errors during the **LDU** decomposition and $\mathbf{b}$ is a vector containing the round-off errors while the right hand side is computed. The solution error can be solved from Eq.(1.16)

$$\Delta\hat{\mathbf{X}}^k - \Delta\mathbf{X}^k = (\mathbf{A}+\mathbf{a})^{-1}(\mathbf{b}-\mathbf{a}\Delta\mathbf{X}^k)$$

This equation clearly indicates that the solution errors are proportional to the round-off errors of the matrix as well as of the right-hand side.

The vector $\mathbf{b}$ is proportional to the right-hand side

$$b_i \sim 2^{1-m} B_i$$

where m is the precision of the machine (e.g., m = 32 for 32 bit machines). If our formulation results in a small right-hand side in magnitude, the error of $\mathbf{b}$ is negligible.

Let the computed **L**, **D** and **U** be represented by $\hat{\mathbf{L}}, \hat{\mathbf{D}}$ and $\hat{\mathbf{U}}$ respectively, we have the following evaluation of errors from the matrix

$$\left|a_{ij}\right| \le N_d 2^{1-m}\left(3\left|A_{ij}\right| + 5\left|\hat{L}_{ik}\right|\left|\hat{D}_{kk}\right|\left|\hat{U}_{kj}\right|\right) + O\left((2^{1-m})^2\right)$$

where N_d is the number of equations. There are three factors that can contribute errors. Two of them are N_d and m. When the number of equations is not so large or the machine precision is very high (e.g., when 64 bit machines are used), the errors can be reduced. However, the two factors are usually unchangeable. The third factor is $\left|\hat{L}_{ik}\right|\left|\hat{D}_{kk}\right|\left|\hat{U}_{kj}\right|$. If the matrix is diagonal dominant as is usually encountered in structural analyses, it is the same order of $\left|A_{ij}\right|$. In fluid flow problems, this factor could become very large if small pivots are used. Small pivots could be encountered if the matrix structure or the so-called symbolic ordering is not

suitable for the problem or the matrix is truly ill conditioned. ADINA-F has been optimized to generate the matrix structure such that no artificial ill conditionings may occur. Therefore the small pivots usually come from the ill conditioned matrix.

The conditioning of the matrix is unknown unless we invert the matrix and evaluate the norms of the two matrices. But if we had inversed the matrix, we would not need to know the condition. ADINA-F prints out some useful information during the **LDU** decomposition (in SKYLINE and sparse solvers) or during the incomplete Cholesky factorization (in iterative solvers), which can be used to predict the conditioning of the matrix. They are

$A_{\min}, I_{\min}$ = minimum unfactorized diagonal and the corresponding equation number defined as $|A_{\min}| = |A_{I_{\min} I_{\min}}| = \min\{|A_{ii}|\}$.

$A_{\max}, I_{\max}$ = maximum unfactorized diagonal and the corresponding equation number defined as $|A_{\max}| = |A_{I_{\max} I_{\max}}| = \max\{|A_{ii}|\}$.

$D_{\min}, I_{\min}$ = minimum factorized diagonal and the corresponding equation number defined as $|D_{\min}| = |D_{I_{\min} I_{\min}}| = \min\{|D_{ii}|\}$.

$D_{\max}, I_{\max}$ = maximum factorized diagonal and the corresponding equation number defined as $|D_{\max}| = |D_{I_{\max} I_{\max}}| = \max\{|D_{ii}|\}$.

The conditioning of the matrix can then be predicted by

$$\frac{D_{\max}}{D_{\min}} \sim \|\mathbf{A}\| \|\mathbf{A}^{-1}\|$$

Hence, it must be the objective to have "good" initial conditions for the solution at every time step, to improve the matrix conditioning and to minimize the magnitude of the right-hand side. Some efficient methods for these purposes are described in the next sections.

13.3.1 Incremental solution procedure

When the initial condition is far from the final solution, the problem can be solved in a few time steps, such that the solutions at any two consecutive time steps are close enough to satisfy the convergence condition in the Newton-Raphson iteration method.

Consider the example of a forced convection problem as shown in Fig.13.20, where the bottom wall is heated. The steady state solution is required.

When the open boundaries are closed, it is the well-known Rayleigh-Benard convection problem that is very unstable and has multiple solutions when the Rayleigh number is larger than a critical value. The velocity that is prescribed at the inlet forces the fluid to flow through the cavity and the heat is therefore transported out of the domain primarily by convection. The forced fluid boundary condition is thus a stabilizing factor in this model compared with the Rayleigh-Benard convection problem. However, the high Reynolds number presents a possible difficulty as well when only the fluid flow is solved.

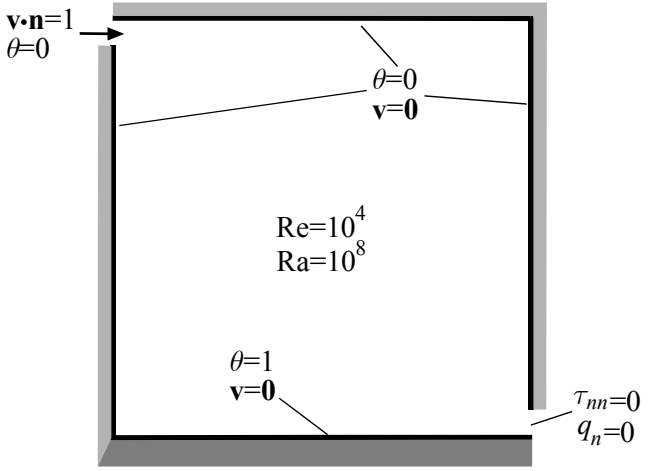


Figure 13.20 Forced convection in a cavity heated at bottom

We may perform the computation in two stages: (1) compute the fluid flow without the influence of heat transfer; (2) compute the heat transfer

with the fluid flow. In each stage, we may achieve the convergence in a few time steps. The fluid flow is computed first because it is a stabilizing factor in the whole model. At the same time, we take the hydrostatic pressure out of the model (see chapter 2 for details regarding this issue).

In stage 1, we execute two time steps. In the first time step, we specify conditions such that the Reynolds number is 10^3, which is an easy problem to solve, so the zero initial condition is fine in the Newton-Raphson iteration. The solution obtained in step 1 is used as the initial condition for time step 2, where we prescribe the full velocity load at the inlet and the Reynolds number becomes the required one, $\mathrm{Re} = 10^4$. During this stage, the bottom temperature is kept as zero, so the Rayleigh number is always zero (see the figure below).

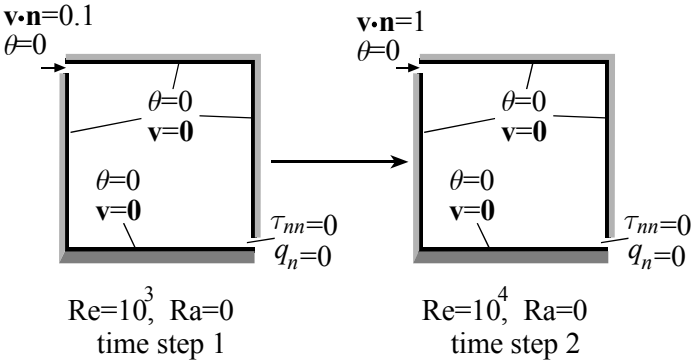


Figure 13.21 Computation in stage 1 for the forced convection problem

In stage 2, we keep the fluid boundary conditions unchanged, while slowly heating the bottom wall. This may be accomplished in a few time steps as well, say two to six. Usually the greatest difficulty in converging in the iteration occurs when the solution changes dramatically. It is effective to impose, in the first step of this stage, a condition of a small Rayleigh number, say $\mathrm{Ra} = 10^4$, with the initial conditions obtained at the end of the stage 1. Then the heating can be increased rapidly. Of course, the solution obtained at every time step is automatically used as the initial condition for the next time step.

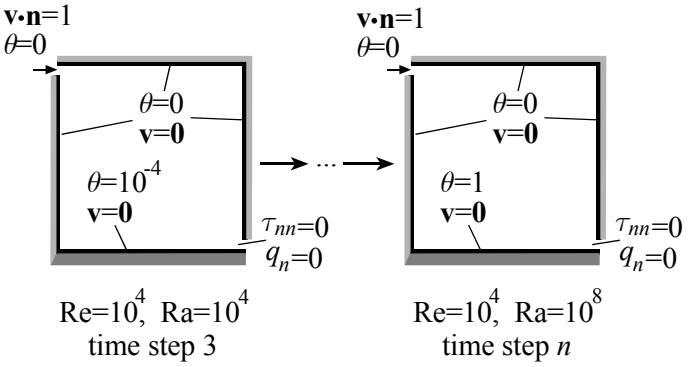


Figure 13.22 Computation in stage 2 for the forced convection problem

To achieve the described computational procedure, we may use time-dependent materials, perform restart runs, etc. Here we use time-dependent boundary conditions. First, we define the inlet velocity and the bottom temperature:

$$\begin{aligned} v &= \overline{v}\, f_1(t) \quad (\overline{v} = 1) \\ \theta &= \overline{\theta}\, f_2(t) \quad (\overline{\theta} = 1) \end{aligned} \tag{1.17}$$

where the time functions are defined as shown in the figure below.

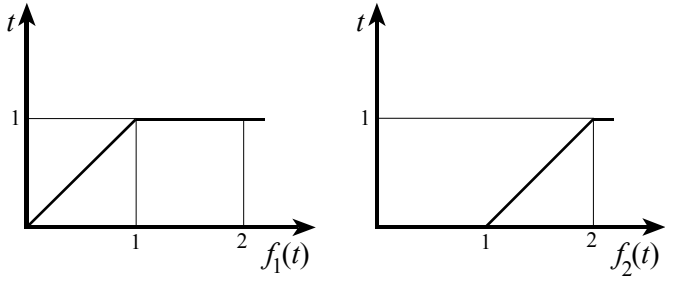


Figure 13.23 Time functions used for the forced convection problem

Secondly, we choose the time steps to obtain the solution (also shown here are the corresponding inlet velocity and the bottom temperature computed according to Eq.(1.17)):

time step	time step length	time at the end of the time step	inlet velocity	bottom temperature
1	0.1	t_1=0.1	0.1	0
2	0.9	t_2=1	1	0
3	0.0001	t_3 = 1.0001	1	0.0001
……	……	……		
n	2-t_{n-1}	t_n = 2	1	1

Note that a steady-state analysis is performed; so the solution obtained at the end of a time step is the steady-state solution under the various conditions at the times considered. Remember that time has no physical meaning in the model; it has merely been used as a parameter to control the boundary conditions.

13.3.2 Improving initial conditions using restart runs

The initial condition in a solution provides the initial guess for the first time step solution. In certain cases, this condition is critical for convergence of the iterations. An example of transient fluid-structure interaction analysis is shown in Fig.13.24, where a water pipe is subjected to small excitation force acting at the bottom. In this example, we have assumed that the gravity force is far larger than the excitation force. If we were to start from the zero velocity condition and a straight pipe to calculate the dynamic response, the solution would diverge (because of the poor initial guess of the final solution). Let d_g and d_F be the deformation scales of the pipe due to gravity and the excitation force, respectively, we expect that $d_F \ll d_g$.

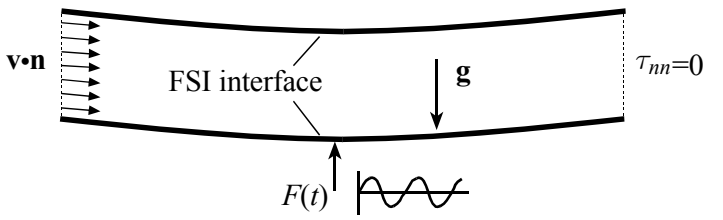


Figure 13.24 Water pipe subjected to a small periodic excitation force

An effective solution will be obtained in two stages. In the first stage, a steady-state analysis of the fluid and structure is performed solely under the gravitational force. In the second stage, a transient analysis of the response due to the excitation force is performed using the solution obtained in stage 1 as an initial condition. This procedure follows closely the actual physics, and is also important for the numerical computations. If the gravity and the excitation force are added simultaneously to the model, the excitation force will be negligible and the pipe will hardly reach the deformation scale d_g because of the small time steps needed in the transient analysis.

We can achieve this result by either of the two methods:

(1) Use a very large time step in the first time step, during which only the gravity force is added. The steady-state solution corresponding to only the gravitational force will be obtained at the end of this time step. Then, reasonable time step lengths are used for the real transient analysis in which the excitation force becomes active. These steps are performed to simulate the dynamic response due to the excitation force. This method is simple and can be performed in one execution. The drawback of this method is that the first time step length is arbitrary. The transient analysis starting from an arbitrary large time step "looks" unnatural as well.

(2) Perform two runs. The first run performs the steady-state analysis with only the gravity force present. The second run performs the transient analysis, based on the initial condition obtained in the first run.

13.3.3 Improving the initial conditions using the mapping file

A solution can always be used as the initial condition for another model as long as the computational domain of the present model lies inside the domain of the earlier model analyzed and a mapping file was saved for the first model. If the two solutions are similar, it serves as a good initial condition.

An advantage of this procedure is that the elements and other conditions of the two runs are independent as well.

13.3.4 Improving the matrix conditioning using the CFL option

The conditioning of the matrix $\mathbf{A}$ in the Newton-Raphson iteration also plays an important role. The matrix could be ill-conditioned due to physical phenomena. This usually happens in high Reynolds number flows. The conditioning can also be bad if the element sizes vary dramatically in the computational domain. When an improper system of units is used, the conditioning of the matrix could be poor as well (see next section in this case). A simple solution is to add a relaxation factor in the Newton-Raphson method

$$\Delta\mathbf{X} = (\mathbf{A} + \lambda^{-1}\mathbf{I})^{-1}\mathbf{B}$$

where λ is the relaxation factor. The factor changes the values of the diagonal elements, thus improving the conditioning of the matrix. When λ is small, it not only improves the matrix conditioning, but also reduces the requirement to have good initial conditions. However, a small λ also slows down the convergence rate in the Newton-Raphson method. There is no universal factor that can be used for all cases. Obviously, the best factor depends on the flow pattern, element sizes, etc. Furthermore, this relaxation factor is not dimensionless. Therefore, λ used in a continuity equation may not be good for the momentum equations, etc.

The automatic time-stepping CFL option in ADINA-F provides an effective tool for improving the matrix conditioning. We have

$$\Delta\mathbf{X} = (\mathbf{A} + \lambda_{\text{CFL}}^{-1}\mathbf{D})^{-1}\mathbf{B}$$

where **D** is a matrix close to a diagonal matrix. It is formulated during the time discretization, on element sizes, velocities, viscosity, etc. More importantly, the CFL number λ_{CFL} is dimensionless and its magnitude can be reasonably predicted from the stability analysis of explicit time integrations. Its lower limit is 1 (the critical value for explicit methods) and its upper limit is infinity when the matrix conditioning is very good. A range of 10^2 to 10^7 could be an initially good choice for many cases.

13.3.5 Improving the matrix conditioning using proper units

An improper system of units can also result in a bad matrix conditioning. For example, if we have a matrix system

$$\mathbf{AX} \equiv \begin{bmatrix} 10^{-3} & 10^{3} \\ 10^{-3} & 0 \end{bmatrix} \begin{bmatrix} v[\text{m/s}] \\ p[\text{Pa}] \end{bmatrix} = \begin{bmatrix} ... \\ ... \end{bmatrix}$$

that is based on the velocity and pressure units as shown in the equation. The conditioning of the matrix is

$$\left\|\mathbf{A}\right\|_{\infty} \left\|\mathbf{A}^{-1}\right\|_{\infty} = 10^{3}(10^{3} + 10^{-3})$$

If we only change meters to millimeters, we obtain a different matrix system that reads

$$\widehat{\mathbf{A}}\widehat{\mathbf{X}} \equiv \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} v[\text{mm/s}] \\ p[\text{kg/mm-s}^2] \end{bmatrix} = \begin{bmatrix} ... \\ ... \end{bmatrix}$$

and the conditioning of the matrix becomes

$$\left\|\widehat{\mathbf{A}}\right\|_{\infty} \left\|\widehat{\mathbf{A}}^{-1}\right\|_{\infty} = 2$$

This procedure can be looked at from another point of view. If we use a length scale $L = 10^3$ and velocity scale $V = 10^3$ in the procedure to automatically nondimensionalize the variables, as available in ADINA-F, we will achieve the same conditioning. Therefore, the task to choose a

proper unit can be accomplished by choosing proper scales for nondimensionalization.

In order to obtain a general guideline for choosing the scales for nondimensionalization, let us consider a model case for incompressible flows

$$\mathbf{AX} \equiv \begin{bmatrix} a & c \\ b & 0 \end{bmatrix} \begin{bmatrix} v \\ p \end{bmatrix} = \begin{bmatrix} ... \\ ... \end{bmatrix}$$

where we have

$$a \sim \mu \Delta x^{d-2} \eta, \quad \left(\eta = 1 + \frac{\rho \Delta x^2}{\mu \Delta t} + \frac{\rho v \Delta x}{\mu} \right)$$
$$b \sim \rho \Delta x^{d-1}, \quad c \sim \Delta x^{d-1}$$

where d is 2 or 3 indicating a two dimensional or three dimensional problem. The **LDU** decomposition is

$$\mathbf{A} = \mathbf{LDU} = \begin{bmatrix} 1 & 0 \\ \frac{b}{a} & 1 \end{bmatrix} \begin{bmatrix} a & 0 \\ 0 & -\frac{bc}{a} \end{bmatrix} \begin{bmatrix} 1 & \frac{c}{a} \\ 0 & 1 \end{bmatrix}$$

Let us apply the scales D^*, L^* and V^* to the model. We determine these scales from the following conditions

$$\tilde{b}\tilde{c} = 1, \quad \frac{\tilde{a}_{\min} \tilde{a}_{\max}}{(\tilde{b}\tilde{c}/\tilde{a})_{\min} (\tilde{b}\tilde{c}/\tilde{a})_{\max}} = 1$$

The result will then be used for general incompressible flows, where the element sizes, velocity magnitudes, etc. vary.

The final scales that can be used are

$$D^* = \rho, \quad L^* = \overline{\Delta x}, \quad V^* = \frac{\mu \overline{\eta}}{\rho \overline{\Delta x}}$$

where

$$\overline{\Delta x} = \sqrt{\Delta x_{\min} \Delta x_{\max}}, \quad \overline{\eta} = \sqrt{\eta_{\min} \eta_{\max}}$$
$$\eta_{\min} = 1 + \frac{\rho \Delta x_{\min}^2}{\mu \Delta t}, \quad \eta_{\max} = 1 + \frac{\rho \Delta x_{\max}^2}{\mu \Delta t} + \mathrm{Re}_{\Delta}$$

with Re_{Δ} representing the maximum cell Reynolds number.

On the other hand, when temperature is included in the computations, obvious choices of the related scales are

$$C^* = C_v, \quad T^* = (\Delta\theta)_{\max}$$

The temperature datum is a very important factor in minimizing the magnitude of the right-hand side when there are radiation or specular radiation boundary conditions in the model. Recall that the heat flux of these boundary conditions is proportional to

$$q_n \sim (\theta_e^4 - \theta^4) \sim (\theta_e - \theta) \tag{1.18}$$

If the environmental temperature is high, as is usually the case in radiation heat transfer and the calculated temperature is low, as is usually the case in the first time step, Eq.(1.18) leaves a term of extremely large magnitude on the right-hand side of the equations. This difficulty can be avoided by choosing the temperature datum

$$\theta_r^* = \theta_e$$

In this case, the radiation heat flux will be

$$q_n \sim \tilde{\theta}$$

This term will be absorbed in the matrix, leaving zero on the right-hand side.

In compressible flows, ADINA-F forces the temperature scale and datum as

$$T^* = \frac{V^{*2}}{C^*}, \quad \theta_r^* = 0$$

to keep consistency of the state equation, regardless of the input.

In the direct computing of FSI two-way coupling problems, the matrix conditioning is worse than in individual fluid and solid models, not only because more equations are solved, but also because of the large difference in magnitude of the matrix elements. A typical diagonal term in the matrix of the solid region is related to Young's modulus that could be as high as 10^{10} in the unit systems used. A typical diagonal term in the matrix of the fluid region is related to the viscosity that could be as low as 10^{-6} in the same unit system. Therefore, a proper choice of nondimensionalization is even more important in this case.

13.4 Consistent units, SI units and USCS units

ADINA-F solves the fluid equations based on Newton's second law expressed in the form

$$F = Ma$$

where F, M and a are force, mass and acceleration, respectively. In the case of compressible flows, the state equations in the form of the ideal gas law are used. These two equations accept any consistent units. However, there is sometimes confusion with some popular USCS (U.S. Customary System) units, like pound-force (lb_f) and pound-mass (lb_m). We want to discuss briefly on the use of consistent units.

In order to keep the units consistent for all data input for a model, one must determine the primary dimensions. Any variable and parameter that is not in the primary unit must be derived from these units. We show two consistent units here as examples.

The first example starts with the primary units defined in the (automatic) nondimensionalization procedure in fluid model: length (L), velocity (V), density (D) and temperature (T). Other variables and parameters are then derived from the primary units. The following table

gives the expressions of generally consistent units and typical examples in SI and USCS units. In SI units, we use $(L,V,D,T)=(\mathrm{m},\mathrm{m/s},\mathrm{kg/m^3},\mathrm{K})$ and in USCS units, we use $(L,V,D,T)=(\mathrm{in},\mathrm{in/s},\mathrm{lb_m/in^3},{}^{\circ}\mathrm{R})$.

Table 13-1 Consistent units derived from the primary units: length, velocity, density and temperature

Variables	Units	SI units	USCS units
x	L	m	in
v	V	m/s	in/s
ρ	D	$\mathrm{kg/m^3}$	$\mathrm{lb_m/in^3}$
θ	T	K	${}^{\circ}\mathrm{R}$
C_p, C_v	V^2/T	$\mathrm{m^2/K\text{-}s^2}$	$\mathrm{in^2/{}^{\circ}R\text{-}s^2}$
t	L/V	s	s
p	DV^2	$\mathrm{kg/m\text{-}s^2}$ ($\equiv$ Pa)	$\mathrm{lb_m/in\text{-}s^2}$
μ	DVL	kg/m-s	$\mathrm{lb_m/in\text{-}s}$
k	DLV^3/T	$\mathrm{kg\text{-}m/K\text{-}s^3}$	$\mathrm{lb_m\text{-}in/{}^{\circ}R\text{-}s^3}$
F	DV^2L^2	$\mathrm{kg\text{-}m/s^2}$ ($\equiv$ N)	$\mathrm{lb_m\text{-}in/s^2}$
e	V^2	$\mathrm{m^2/s^2}$	$\mathrm{in^2/s^2}$

The second example starts with the primary units length (L), time (t), force (F) and temperature (T). Other variables and parameters are then derived from the primary units. The following table gives the expressions of generally consistent units and typical examples in SI and USCS units. In SI units, we use $(L,t,F,T)=(\mathrm{cm},\mathrm{s},\mathrm{dyne},\mathrm{K})$ and in USCS units, we use $(L,t,F,T)=(\mathrm{ft},\mathrm{s},\mathrm{lb}_f,{}^{\circ}\mathrm{R})$.

Table 13-2 Consistent units derived from the primary units: length, time, force and temperature

Variables	Units	SI units	USCS units
x	L	cm	ft
t	t	s	s
F	F	dyne	lb_f
θ	T	K	$^\circ$R
ρ	Ft^2/L^4	g/cm^3	$\text{lb}_f\text{-s}^2/\text{ft}^4$ ($\equiv \text{slug/ft}^3$)
v	L/t	cm/s	ft/s
C_p, C_v	$L^2/(t^2T)$	$\text{cm}^2/\text{K-s}^2$	$\text{ft}^2/^\circ\text{R-s}^2$
p	F/L^2	dyne/cm^2	lb_f/ft^2 ($\equiv$ psf)
μ	Ft/L^2	g/cm-s	$\text{lb}_f\text{-s}/\text{ft}^2$ ($\equiv$ slug/ft-s)
k	$F/(Tt)$	g-cm/K-s^3	$\text{lb}_f/^\circ\text{R-s}$ ($\equiv \text{slug-ft}/^\circ\text{R-s}^3$)
e	L^2/t^2	cm^2/s^2	ft^2/s^2

The key in keeping units consistent is to always work with the primary units rather than conventional units. For example, in the second example above, the density is in the unit $\text{lb}_f\text{-s}^2/\text{ft}^4$, not in the unit lb_m/ft^3.

Another common approach is to work using nondimensionalized units. That is, all data input has been made dimensionless. This method is popular with scientific researchers. This approach then uses the Reynolds number, Prandtl number, etc. in the definition of the problem to be solved.

13.5 Use of memory and disk

13.5.1 Memory usage

The memory used in ADINA-F falls into two categories: the basic memory used to store the data input and generated data (including the

matrix system) and additional memory that is required by the specific solver selected. The memory that ADINA-F outputs is in words. One word equals 4 bytes and 8 bytes, respectively, on 32 and 64 bit machines. In the following descriptions, a 32-bit machine is assumed.

(2) Basic memory

This part includes the memory for the solution variables, element group data, boundary conditions, etc. The major portion of the memory is of course used to store the matrix system. To ease the explanation of the memory usage here, we assume that these data are in-core as well. Although the element group data and the matrix may be out-of-core, the in-core solution, if it can be performed, is generally more effective.

The basic memory used can be divided into two portions

$$m = m_1 + m_2$$

where m_1 and m_2 are the memory for preliminary storage and for storing and manipulating the fluid matrix. The amount needed can be predicted by

$$m_1 = \alpha_n N_n + \alpha_e N_e \qquad \text{words}$$
$$m_2 = 5\alpha_m n_d^2 N_n = 5\alpha_m n_d N_d \qquad \text{words}$$

where N_n is the number of nodes, N_e is the number of elements, N_d ($\approx n_d N_n$) is the number of equations (or called the number of degrees of freedom), n_d is the number of solution variables, and the α's are factors. α_n and α_e vary in the ranges

$$\alpha_n \sim 25\text{-}60, \quad \alpha_e \sim 20\text{-}25$$

depending on the problem type. The factors that may increase the use of memory are, for example, three-dimensional flows, including temperature degrees of freedom, moving mesh conditions, turbulence models, etc. The factor n_d varies in the range

$$n_d \sim 1\text{-}7$$

corresponding to the number of variables $(u, v, w, p, \theta, K, \varepsilon/\omega)$ included in the model. It is 1 if temperature is the only variable and 7 if three-dimensional turbulent flows including heat transfer are computed. Except the solver SKYLINE, all solvers store the matrix in a compact form. Assuming that either the sparse solver or iterative solver is used, the factor α_m depends then only on the type and the connectivity of the elements

$$\alpha_m \approx \begin{cases} 8 & \text{in triangular 3-node element} \\ 25 & \text{in quadrilateral 9-node element} \\ 30 & \text{in tetrahedral 4-node element} \\ 125 & \text{in brick 27-node element} \end{cases} \tag{1.19}$$

It is clear that, for most realistic problems, $m_2 \gg m_1$. Therefore, it is reasonable to keep all data corresponding to the memory m_1 in-core.

The direct solver SKYLINE and the iterative solvers RPBCG and RPGMRES automatically use the hard disk (out-of-core) once the memory m_2 is not sufficient. However, the speed of solution can then be much reduced. If the sparse solver is selected, the initially assembled matrix must be in core.

(2) Memory used in solvers

This part includes the memory used in the procedure to solve the equations. All iterative solvers do not require additional memory. However, if the sparse solver is used, additional memory is required. In the sparse solver, the original matrix is expanded to a new matrix whose size depends on the fill-ins to the original matrix. This part of the memory is highly dynamic and considered separately

$$M \sim \alpha_M m_2$$

where the factor α_M would be typically 2 to10. This part of the memory can be allocated out-of-core (using disk space) if the memory is not sufficient and the out-of-core sparse solver is selected. The advantage of the out-of-core sparse solver is that it does not depend much on the memory

assigned to it. Usually the out-of-core sparse solution takes twice the amount of time as the in-core.

Regarding the memory usage in ADINA-F, we recommend the following strategies to manage the memory efficiently:

- If the computer has sufficient memory to hold both m and M in-core, you need only to specify the memory m. The sparse solver, in case it is used, will dynamically allocate the actual memory as required during the procedure to solve the equations. If there is not enough memory available, it will print out a message that indicates lack of memory and the program stops.
- If the computer has only sufficient memory for m, we have to use the out-of-core sparse solver. In this case, the memory M must be explicitly specified. The sparse solver will then manage to work in-core if the memory M is sufficient and go out-of-core otherwise. Of course, M must be less than the difference between the available memory and m.
- If the available memory is not large enough to hold the required m, the sparse solver cannot be used. The out-of-core iterative solvers must be used. Only the RPBCG and RPGMRES out-of-core solvers are available. No special input is required to have these solvers work out-of-core. They automatically use the disk space whenever the memory m is not large enough. However, the speed of solution highly depends on the blocks of the sub-matrices that are generated. The less memory available, the larger the solution time.

The above prediction is only for the most frequently encountered cases. Certain cases need further comments.

- When mass transfer and moving meshes are included, the memory for the solver is divided into three parts

$$M = M_1 + M_2 + M_3$$

where M_1, M_2 and M_3 are the memory assigned to the fluid equations, mass transfer equations and moving mesh equations, respectively. They are determined according to the ratios $N_d : N_n : N_n/2$. Then, if the out-of-core sparse solver is used for the

fluid, the solvers for the mass ratios and moving meshes may or may not be in-core.

- In iterative computing of two-way fluid-structure coupled problems and in direct computing of the one-way fluid-structure coupled problems, the same memory M is assigned to the solvers that are used for both the fluid and the solid models. In other words, the total memory used is $m + 2M$.
- In direct computing of two-way fluid-structure coupled problems, the number of nodes, number of elements, and number of equations, etc. are all referring to the coupling system.
- When specular radiation boundary conditions are applied, the view factor matrix and radiosity matrix require storage as well. Usually, this part of storage is much larger than that for the fluid flow equations and can be evaluated as

$$m_3 = 2N_{ns}^2 + 2N_{es}^2 \qquad \text{words}$$

where N_{ns} and N_{es} are the number of specular nodes and elements, respectively. This storage can be assigned to out-of-core memory. The view factor matrix is computed first and stored on disk. It will be read in while the radiosity matrix is assembled. Let n_{bc} and n_{bi} be the number of sub-matrix blocks while the view factor matrix is computed and then read-in, respectively, n_{bi} is generally larger than n_{bc}. Let n_{br} be the number of sub-matrix blocks of the radiosity matrix; the storage required in specular radiosity computations becomes

$$m_3 = \frac{2N_{ns}^2}{n_{bi}} + \frac{2N_{es}^2}{n_{br}}$$

The basic memory is thus modified to

$$m = m_1 + \max\left\{m_2, m_3\right\}$$

The computational cost in specular radiosity calculations is directly related to

$$\text{CPU} \sim O(n_{br}^2, n_{bc}^2)$$

When the assigned storage m is not sufficient to hold the view factor matrix and/or radiosity matrix in-core, ADINA-F automatically works with disk storage. The numbers, n_{bc}, n_{bi} and n_{br} are printed to inform users of the possible computational effort. The allowed maximum number of blocks is 1000. It would be computationally very inefficient to have more blocks.

13.5.2 Disk usage

Some disk space is required dynamically during the execution of ADINA-F. Part of that is occupied by the solution results while another part is temporarily used during the program execution. The major disk storage files are listed in the following table. The disk space requirement is approximately given and may be more or less depending on the type of problem solved. For example, if the fluid solver runs in-core, the matrix is not saved on disk; if the specular radiation boundary conditions is not used, no view factor and radiosity matrix disk storage is required; if the mapping file is not selected, it will not be printed; etc.

The notation used in this table is

N_n = number of nodes
N_e = number of elements
N_t^s = number of time steps that solutions are saved
N_d = number of equations
α_m = the factor as defined in Eq.(1.19)
N_{ns} = maximum number of specular nodes in one group
N_{es} = maximum number of specular elements in one group
n_e = number of nodes per element
n_d = number of variables (1 to 7 for $u, v, w, p, \theta, K, \varepsilon/\omega$)

Table 13-3 Major files used in ADINA CFD & FSI

File name	Explanations	Data stored	File size in words
fort.80 (=*.por)	porthole file	nodal solutions	$20N_nN_t^s$
		element solutions	$44N_eN_t^s$
fort.74 (=*.res)	restart file	nodal solutions	$30N_n$
		element solutions	$12N_e$
fort.75 (=*.tem)	temperature file	temperature solutions	$2N_nN_t^s$
fort.77 (=*.map)	mapping file	nodal coordinates	$6N_n$
		element connectivities	n_eN_e
		nodal solutions	$2N_dN_t^s$
fort.95	temporary file	element group data	$24N_e$
		fluid matrix	$2\alpha_mN_d$
		specular matrix	$2N_{ns}^2$
fort.78	temporary file	specular view factors	$2N_{es}^2$

Note that, in FSI computations, additional disk storage is required in solid models. Please refer to the ADINA Solids & Structures Theory and Modeling Guide for more information.

13.6 Brief summary of recommendations for solutions

The following is a brief summary of some recommendations regarding the use of the CFD capabilities when modeling a physical problem. These recommendations based on the flow to be solved are not exhaustive but merely given to help a user in starting to employ the ADINA CFD solution capabilities.

While we refer to the Re flow conditions, the same comments are applicable to the Pe number flow conditions. Also, while we seem to imply steady-state analyses, the same comments are applicable to transient analyses.

For low Re incompressible flows

The 9-node 2D and 27-node 3D elements can be used, if the elements support the boundary conditions, loading applied, and other features you want to use. The direct sparse solver is used as the default option for solution of the algebraic equations, hence the memory requirement and solution time may be large if the model is large. In particular for 3D problems, the bandwidth may be large. We hardly use the iterative solvers. For FSI problems, the FCBI and FCBI-C elements are better used.

Use the FCBI elements, preferably the 4-node element in 2D and 8-node element in 3D, instead of the triangular and tetrahedral elements (which however are largely used in free-form meshing and must be used in adaptive meshing). With these elements also by default the sparse solver is used, but a smaller bandwidth is encountered. The memory and solution time requirements may also be large when large models are solved, but not as large as when using the 9-node and 27-node elements referred to above. We hardly use the iterative solvers. These elements are recommended for use in FSI problems.

For very large fluid models, the FCBI-C elements are an attractive option. With these elements relatively fine meshes may need to be used. As for the FCBI elements, here too, the 4-node element in 2D and 8-node element in 3D are preferred. Using the FCBI-C elements, the sparse solver can be used, but the default is the algebraic iterative solver, type 1. Using this iterative solver, memory requirements are much lower for large models than using the sparse solver. But for the iteration, it is important to employ tight enough convergence tolerances. In practice, you may solve the algebraic equations using the default tolerances for the iterative solver, study the solution, and if questions arise as to whether a converged solution has been obtained, thereafter continue the iteration to reach a tighter convergence tolerance.

For high Re incompressible flows

Using the 9-node 2D and 27-node 3D elements very fine meshes may be required to have stability, that is, to have element Re numbers smaller than about 6. The elements are formulated using the standard Galerkin method (using the velocity-pressure formulation) without upwinding [1]. Hence these elements are not very suitable for high Re flows.

The FCBI and FCBI-C elements should be used with the comments given above.

Slightly compressible flows

For slightly compressible flows, the same comments as given above are applicable.

High-speed and low-speed compressible flows

For high-speed and low-speed compressible flows, only the triangular 2D and tetrahedral 3D elements are available. The algebraic equations are solved by default using the sparse solver, but for large models the multi-grid solver can be used.

Ref.

[1] KJ Bathe, Finite Element Procedures, Prentice-Hall, 1996.

This page intentionally left blank

Appendix-1 List of figures

Figure 2.1 Rotating coordinate frame 32
Figure 2.2 Coordinate system for axisymmetric flows 35
Figure 2.3 Illustration of porous media models 43
Figure 2.4 A typical example of using ALE coordinates 71
Figure 2.5 Definition of the ALE coordinate system 72
Figure 3.1 A typical boundary condition set for incompressible, slightly compressible and low-speed compressible flows 91
Figure 3.2 A typical boundary condition set for temperature 92
Figure 3.3 Prescribed rotational velocity 95
Figure 3.4 No-slip condition on fixed walls for incompressible, slightly compressible and low-speed compressible flows 99
Figure 3.5 Slip condition on fixed walls for incompressible, slightly compressible and low-speed compressible flows 100
Figure 3.6 Angular velocity boundary condition for incompressible, slightly compressible and low-speed compressible flows 101
Figure 3.7 No-slip condition on moving walls for incompressible, slightly compressible and low-speed compressible flows 102
Figure 3.8 Slip condition on moving walls for incompressible, slightly compressible and low-speed compressible flows 103
Figure 3.9 Moving wall condition (type=tangential) for incompressible, slightly compressible and low-speed compressible flows 105
Figure 3.10 Moving wall condition (type=rotational) for incompressible, slightly compressible and low-speed compressible flows 105
Figure 3.11 Sketch of a free surface condition 107
Figure 3.12 A typical phase-change boundary condition 110
Figure 3.13 A typical gap boundary condition 111
Figure 3.14 Diagram of gap open-close conditions 112
Figure 3.15 2D Rotor-Stator Interaction 113
Figure 3.16 Two Passing Trains/Cars 114
Figure 3.17 Sliding mesh interface coupling 115
Figure 3.18 2-D geometry for sliding mesh interface 115
Figure 3.19 Physical conditions at edge of sliding mesh interface 116
Figure 3.20 A weak rotor-stator interaction 119
Figure 3.21 Two rotors in a chamber 120

Figure 3.22 Model setup of a multiple-reference frame 121
Figure 3.23 A typical convection boundary condition 125
Figure 3.24 Schematic representation of radiation energies corresponding to an incident ray 127
Figure 3.25 A typical usage of specular groups 129
Figure 3.26 A piecewise linear function 147
Figure 4.1 A typical boundary condition set for porous media flows 171
Figure 5.1 A typical boundary condition set for turbulence variables 180
Figure 6.1 Definition of a two-dimensional control volume 201
Figure 6.2 Sub-control volumes and their interfaces within one triangular element 202
Figure 6.3 Sub-control volumes and their interfaces within one tetrahedral element 203
Figure 6.4 Definition of flow regimes in high-speed compressible flows 206
Figure 6.5 Illustration of the supersonic condition at inlet 207
Figure 6.6 Illustration of the subsonic condition at inlet 207
Figure 6.7 Illustration of the subsonic condition at outlet 208
Figure 6.8 Illustration of the supersonic condition at outlet 209
Figure 6.9 A typical boundary condition set for external high-speed compressible flows 210
Figure 6.10 A typical fluid boundary condition set for internal high-speed compressible flows 210
Figure 6.11 No-slip condition on fixed walls for high-speed compressible flows 217
Figure 6.12 Slip condition on fixed walls for high-speed compressible flows 218
Figure 6.13 Angular velocity condition for high-speed compressible flows 219
Figure 6.14 No-slip condition on moving walls for high-speed compressible flows 220
Figure 6.15 Slip condition on moving walls for high-speed compressible flows 221
Figure 6.16 Moving wall condition (type=tangential) for high-speed compressible flows 223
Figure 6.17 Moving wall condition (type=rotational) for high-speed compressible flows 224
Figure 6.18 External flow condition 225
Figure 6.19 Sketch for inlet and outlet flow conditions 226
Figure 9.1 Illustration of fluid-structure interactions 258
Figure 9.2 Coupling of fluid and solid nodes 260
Figure 9.3 Measure of the distance between fluid and solid FSI boundaries 261
Figure 9.4 Interpolation of fluid stresses in time in indirect computing of one-way fluid-structure coupling 270

Figure 9.5 Typical example of O-shape 2D F-S-F connections in fluid-structure interaction analyses. 273
Figure 9.6 Possible methods for generating O-shape 2D fluid elements in F-S-F connections 273
Figure 9.7 Application of fluid-structure interface conditions to beam elements ... 274
Figure 9.8 Typical example of C-shape 2D F-S-F connections in fluid-structure interaction analyses. 275
Figure 9.9 Possible methods for generating C-shape 2D fluid elements in F-S-F connections 275
Figure 9.10 Typical example of O-shape 3D F-S-F connections in fluid-structure interaction analyses. 276
Figure 9.11 Typical example of C-shape 3D F-S-F connections in fluid-structure interaction analyses. 277
Figure 9.12 Illustration of overall fluid-structure interactions. The coupled region is $\Omega_{FSI} = S \bigcup \Omega_p \bigcup \Omega_s$ 282
Figure 10.1 Line elements 287
Figure 10.2 2D/axisymmetric 3-node element 289
Figure 10.3 2D/axisymmetric 9-node element 290
Figure 10.4 2D/axisymmetric 6-node element 292
Figure 10.5 3D 4-node element 293
Figure 10.6 27-node 3D element 295
Figure 10.7 FCBI 2D elements 297
Figure 10.8 FCBI 3D elements 298
Figure 10.9 FCBI-C elements 300
Figure 11-1 Default set of outer relaxation factors in segregated method 313
Figure 11-2 Default set of inner relaxation factors in segregated method 315
Figure 11-3 Ordering in sparse solver 323
Figure 12.1 Prescribe a radial velocity 333
Figure 12.2 Sketch of conjugate heat transfer 336
Figure 12.3 Application of pressure-datum feature to fluid-structure interaction problems. 338
Figure 12.4 A typical case of reaching an unacceptable element. 340
Figure 12.5 Valid mesh with a proper subdivision of the domain 340
Figure 12.6 Definition of the leader-follower options 342
Figure 12.7 Definition of VOF-wall angle 354
Figure 13.1 Typical problems that incompressible flow model can be applied 382
Figure 13.2 Typical problems that slightly compressible flow model can be applied 382

Figure 13.3 Typical problems that low-speed compressible flow model can be applied....383
Figure 13.4 Typical problems that high-speed compressible flow model can be applied....383
Figure 13.5 Typical problems that porous media flow model can be applied....384
Figure 13.6 Example of using porous media flow model....384
Figure 13.7 Time step sizes in periodic solutions....388
Figure 13.8 Time step sizes in wave propagation problems....389
Figure 13.9 Time steps in diffusion problems....390
Figure 13.10 Choosing the computational domain....391
Figure 13.11 Avoid lower dimensional flows in the model....392
Figure 13.12 Using equivalent physical models....392
Figure 13.13 Multiple normal-traction conditions in incompressible flow models 394
Figure 13.14 Multiple normal-traction conditions in compressible and incompressible flow models....394
Figure 13.15 Fixed pressure condition must be applied to confined incompressible flows....395
Figure 13.16 Solution of a circular laminar jet....396
Figure 13.17 Distorted element in a moving boundary problem....402
Figure 13.18 Use of leader-follower option....403
Figure 13.19 Using contact surfaces in FSI analyses....404
Figure 13.20 Forced convection in a cavity heated at bottom....408
Figure 13.21 Computation in stage 1 for the forced convection problem....409
Figure 13.22 Computation in stage 2 for the forced convection problem....410
Figure 13.23 Time functions used for the forced convection problem....410
Figure 13.24 Water pipe subjected to a small periodic excitation force....412

This page intentionally left blank

Appendix-2 List of tables

Table 1-1 Overview of capabilities of ADINA CFD & FSI 26
Table 3-1 Functionalities of incompressible, slightly compressible and low-speed compressible flow models 80
Table 3-2 Transformations of temperature units to absolute temperature units 126
Table 4-1 Functionalities in the porous media flow model 167
Table 5-1 Functionalities in K-ε and K-ω turbulence models for incompressible, slightly compressible and low-speed compressible flows 175
Table 6-1 Functionalities in the high-speed compressible flow model 191
Table 7-1 Functionalities in the K-ε turbulence model for high-speed compressible flows 235
Table 8-1 Functionalities for mass transfer analyses 243
Table 9-1 Elements in connections of fluid and solid models in fluid-structure interaction problems 272
Table 13-1 Consistent units derived from the primary units: length, velocity, density and temperature 418
Table 13-2 Consistent units derived from the primary units: length, time, force and temperature 419
Table 13-3 Major files used in ADINA CFD & FSI 425

Appendix-3 List of error/warning messages

***** CODE ADF1005**:

Too many error/warning messages for code number ADFxxxx have been encountered.

This message indicates that the message of code number ADFxxxx has been encountered more than 10 times, where xxxx is a 4-digit number. Any error/warning message that has been printed 10 times will no longer be printed. Note that in case the program terminates, the cause could be related to ADFxxxx not printed. The message ADFxxxx here represents any message in ADINA-F.

***** CODE ADF1006:**

Too many error/warning messages have been encountered.

This message indicates that an error or warning message has been encountered more than 10 times. Any error/warning message that has been printed 10 times will no longer be printed. Note that in case the program terminates, the cause could be related to any error or warning messages that have been printed more than 10 times.

***** ERROR *** CODE ADF2002**:

Model may be unstable! Ratio 9.19E+19 of largest to smallest diagonal elements in factorized coefficient matrix is greater than 1.10E+19.

When this ratio is too large, machine round-off errors may become severe.

Possible causes are:

* Solution procedure has been divergent;
* Ratio of largest to smallest elements is too large;
* No fixed pressure condition in confined flows;
* Improper units are used in this model. If this is the reason, you may either change the input to another proper unit system or choose proper scales for the automatic nondimensionalization.

This message indicates a large ratio of the factorized diagonal in the matrix **LDU** decomposition or incomplete decomposition. This message is essentially the same as the message of ADF2006, except that the corresponding solution variables here are unable to be identified.

***** ERROR *** CODE ADF2003:**

```
Error encountered while reading data block (code 1111):
  "CURRENT_DATA_ID".
Last input data block is:
  "LAST_DATA_ID".
Data could be wrong some where between the two data blocks.
This could mean an internal error or a wrong data file was used.
```

This message indicates an error encountered in the *.dat file. This file consists of a few data blocks. Each block has a brief identification description. "CURRENT_DATA_ID" is the current block description while "LAST_DATA_ID" is the last block description. The error could be in the current block or the last block. The following data blocks are sequentially stored in the data file:

- MASTER DATA
- TIME STEP BLOCK DATA
- SAVE AND PRINT BLOCK DATA
- TIME FUNTION DATA
- ONE-D FUNCTION DATA
- BULK DENSITY DEFINITION DATA
- SKEW SYSTEM DATA
- NODAL DATA
- CONSTRAINT DATA
- MATERIAL DATA OF ELEMENT GROUP 1
- MASS MATERIAL DATA OF ELEMENT GROUP 1
- ELEMENT GROUP DATA OF ELEMENT GROUP 1
- GEOMETRY AND BCD HEAD DATA
- GEOMETRY DATA
- BOUNDARY CONDITION HEAD DATA
- BOUNDARY CONDITION DATA
- HEAD OF BOUNDARY NODE LIST DATA
- BOUNDARY NODE LIST DATA
- 2D NORMAL-TRACTION LOAD DATA
- 2D HEAT FLUX LOAD DATA
- 3D NORMAL-TRACTION LOAD DATA
- 3D HEAT FLUX LOAD DATA
- PRESCRIBED LOAD DATA
- CONCENTRATE LOAD DATA

- 2D MASS FLUX LOAD DATA
- 3D MASS FLUX LOAD DATA
- PRESCRIBED MASS LOAD DATA
- ROTATIONAL LOAD DATA
- MASS INITIAL CONDITION DATA
- HEAD OF FLUID INITIAL CONDITION DATA
- FLUID INITIAL CONDITION DATA

When this message is encountered, first make sure a correct data file is used. Refer to the suggestions for message ADF2111.

In case of an error that you may have made during a manual modification of the data file, the last and the current data block information indicates where the error could be.

*** WARNING *** CODE ADF2004:

Diagonal element 9.99E-21 of factorized effective stiffness matrix is smaller than 1.99E-20. A constant pivot 1.00E+20 has been assigned to equation 430000, which corresponds to PRESSURE at node (label) 99999.

Possible causes are:

* Incorrect boundary conditions have been applied;
* Physical model is unstable;
* Solution procedure is divergent.

This message indicates a zero pivot encountered during the **LDU** decomposition or incomplete decomposition. The corresponding solution variable is also identified.

If this message appears in the first iteration in the first time step, a direct possibility is that the model is ill posed.

If the corresponding solution variable is pressure, make sure that

- the pressure degree of freedom is active.
- the required fixed pressure condition (e.g., in confined flows) is applied.
- the viscosity is correctly specified.
- no velocities are wrongly fixed (e.g. inside the computational domain).
- the transient analysis or a proper CFL number has been used in Euler high-speed flows.
- a proper unit system is used.

If the corresponding solution variable is velocity, make sure that

- at least one wall or its analogue is applied.
- the viscosity is correctly specified.
- a proper unit system is used.

If the corresponding solution variable is temperature, make sure that

- at least one prescribed temperature condition or its analogue (convection, radiation, etc.) is applied if a steady-state analysis is performed. If the only analogue of prescribed condition is a convection boundary condition, the convection coefficient $\bar{h}(t)$ must be nonzero. The associated time function must be checked as well. Similarly, if the only analogue of prescribed condition is a radiation boundary condition, the Stefan-Boltzmann constant σ, the shape factor f and the emissivity e must be nonzero.
- the heat conductivity is correctly specified.

If the corresponding solution variable is a turbulent variable, make sure that

- at least one prescribed condition has been applied to that variable. A wall condition is considered as a prescribed condition as well.
- the laminar viscosity is correctly specified.

If the corresponding solution variable is related to moving mesh, the equation number is the same as the nodal number. Make sure that a valid mesh has been used, particularly near the node where the singularity appears.

If this message appears after a few iterations, the iteration may be divergent. It could be an indication that the model is improperly defined, load increments are too large, the physical phenomenon is unstable, etc. Refer to the topic "Model preparation and testing" for helpful tips.

***** ERROR/WARNING *** CODE ADF2005:**

Iteration limit is reached without convergence.
Equilibrium is not established from time: 1.23450E-21
to time: 2.23449E-21

Here are some suggestions for improving convergence:

* Resolve all error/warning messages.
* Use proper units to avoid machine round-off errors.
* Apply a proper force or displacement relaxation factor in fluid-structure interaction analyses.
* Use a proper CFL number in automatic time stepping method.
* In transient analyses, use smaller time steps.
* In steady-state analyses, incrementally apply loads (prescribed velocities, normal-tractions, etc.) step-by-step, or/and use time-dependent materials, starting with easier material values and eventually approaching to the real values.
* In transient analyses of fluid-structure interactions, a good initial condition of fluid variable is crucial. One can achieve that by using a very large fictitious first time step, which provides a good initial condition for the really transient analyses starting at second time step. One can also achieve that by using a restart run, in which the (steady-state) result of first run provides initial conditions for the (transient) analysis in the second run.

This message indicates unsuccessful equilibrium iterations in the current time step. If the automatic time step (ATS option) is selected, the program will continue using a smaller time step.

Many possibilities could cause the divergence. Besides the suggestions shown in this message, the topic "Model preparation and testing" contains more detail and helpful tips. If the iteration residuals show a converging tendency, you may increase the number of iterations that is permitted to reach a converged solution.

***** WARNING *** CODE ADF2006:**

Model may be unstable! Ratio 1.2339E-21 of largest to smallest diagonal elements in factorized coefficient matrix is greater than 1.E+11. The largest diagonal element -2.2339E-21 corresponds to Y-VELOCITY at node (label) 999999 and the smallest diagonal element 3.23449E-21 corresponds to PRESSURE at node (label) 88888888.

When this ratio is too large, machine round-off errors may become so severe that stable models become unstable.

Possible causes are:

* Solution procedure has been divergent;
* Ratio of largest to smallest elements is too large;
* No fixed pressure condition in enclosures;
* Improper units are used in this model. If this is the reason, you may either change the input data in another unit system or use the automatically nondimensional procedure by choosing proper scales.

This message indicates a large ratio of the factorized diagonal in the matrix **LDU** decomposition or incomplete decomposition. The corresponding solution variables, nodal numbers and equation number are also identified. A large ratio usually indicates a poor conditioning of the matrix. If the minimum factorized diagonal is zero or extremely small (say smaller than 10^{-20}), this message is essentially the same as the message of ADF2004. In this case, check the explanations for that message.

Certain factors can contribute to the ill conditioning:

- the problem is ill posed.
- an improper unit system is used.
- element sizes vary too much.
- highly nonlinear problems (non-Newtonian fluid flows, high-Reynolds/Peclet/Rayleigh number flows, etc.).

Refer to the topic "Model preparation and testing" for helpful tips to exclude the possibility of ill posed problem. Usually, the matrix conditioning can be improved by choosing a proper unit system. It can also be done in the ADINA-F automatically nondimensional procedure by providing proper scales and reference datums. However, since the problem is nonlinear and the element sizes are likely non-uniform, this improvement has its limit. In these cases, the warning message should be ignored.

***** ERROR *** CODE ADF2007:**

Turbulence model is incompatible with the material model.
Turbulence K-Epsilon material model can only be used for Turbulence K-Epsilon fluid model.

The message indicates that a turbulence K-ε material is used in an element group while the fluid model is not turbulence K-ε. Refer to the related chapters to see available materials.

***** ERROR *** CODE ADF2008:**

Turbulence model is incompatible with the material model.
Turbulence K-Omega material model can only be used for Turbulence K-Omega fluid model.

The message indicates that a turbulence K-ω material model is used in an element group while the fluid model is not turbulence K-ω. Refer to the related chapters to see the available materials.

***** ERROR *** CODE ADF2010:**

Incorrect ADINA-F data file. This could occur if:

* The file is not an ADINA-F data file;
* The version of the data file does not match the program version;
* There is no permission to open the data file.

The message indicates that data file for the fluid model is not recognizable.

When this message is encountered, first make sure a correct data file is used. Make sure the version of AUI used to generate the data file is consistent with the version of ADINA-F. It is best to use the latest version. If the *.dat file was generated using an earlier version of AUI, regenerate it using the consistent version.

If the data file was generated using another machine, make sure the data is portable. In general, use a formatted (ASCII) data file if the AUI and ADINA-F are installed in different machines, although most Unix machines can use binary data files interchangeably.

If you have modified the *.dat file, make sure that the format in the modification is correct. It is not recommended (although not restricted) to edit some parameters in the data file.

Check also if the file has been protected so that it is not readable.

***** ERROR *** CODE ADF2011:**

Incorrect ADINA data file. This could occur if:

* The file is not an ADINA data file;
* The version of the data file does not match the program version;
* There is no permission to open the data file.

This message is essentially the same as message of ADF2010, except that the data file corresponds to the solid model. Only in formulation of fluid-structure interactions, a solid data file is required. In addition to the explanations made for the message ADF2010, the consistency between the versions of ADINA and ADINA-F must also be checked.

***** ERROR *** CODE ADF2012:**

File name LONG_FILE_NAME is too long.

This message indicates that a file name is too long (more than 128 characters). Avoid using long names for your files.

***** ERROR *** CODE ADF2013:**

Coarsening in multi-grid ended at level 11 since all inter-block links are too weak.

This message indicates that a poor condition has been encountered in multi-grid solver. Change the solver to the sparse solver or other iterative solvers.

***** ERROR *** CODE ADF2014:**

Multi-grid levels have been reduced to 11 due to memory limit.

This message indicates that multi-grid solver could not be working properly because of the memory limitation. Change the solver to the sparse solver or other iterative solvers.

***** ERROR *** CODE ADF2015:**

Failed in multi-grid solver: status = "status_WHEN_ERROR_OCCURS".

This message indicates that multi-grid solver is unable to converge. Change the solver to the sparse solver or other iterative solvers.

***** ERROR *** CODE ADF2016:**

No enough memory to continue. Currently available status is: "status_WHEN_ERROR_OCCURS".

This message indicates that the memory assigned to ADINA-F is not sufficient for the multi-grid solver. Usually other solvers may not work as well because the memory limitation. More memory is required.

***** ERROR *** CODE ADF2017:**

Incorrect data passed out from the user-supplied boundary condition routine. B(I,VARIABLE_NAME) must be zero since there is no degree of freedom for that variable.

This message indicates that an error has been found in the output of the user-supplied boundary condition subroutine. Refer to the description on how to use user-supplied boundary condition.

A user-supplied boundary condition is defined as a source term that is added to the right hand side of the equation. This source is expressed as $A + B_i\phi_i$, where ϕ_i are all possible solution variables. A nonzero B_i must corresponds to an active variable ϕ_i.

***** WARNING *** CODE ADF2018:**

Incorrect data passed out from the user-supplied boundary condition routine. B(I,VARIABLE_NAME) = 1.230E+19 is too large.

This message indicates that a possible error has been found in the output of the user-supplied boundary condition subroutine. Refer to the description on how to use user-supplied boundary condition.

A user-supplied boundary condition is defined as a source term that is added to the right hand side of the equation. This source is expressed as $A + B_i\phi_i$, where ϕ_i are all possible solution variables. A very large B_i usually indicates a non-initialized number.

***** WARNING *** CODE ADF2019:**

Improper data passed out from the user-supplied boundary condition

routine. B(22) = 1.230E+19 may cause instability. Negative value should be assigned.

This message indicates that a possible error has been found in the output of the user-supplied boundary condition subroutine. Refer to the description on how to use user-supplied boundary condition.

A user-supplied boundary condition is defined as a source term that is added to the right hand side of the equation. This source is expressed as $A + B_i\phi_i$, where ϕ_i are all possible solution variables. A positive B_i could cause numerical instability. Rewrite the source term such that the coefficient is non-positive. For example, $A' = A + (1+\xi)B_i\phi_i$ and $B_i' = -\xi B_i$, where ξ is a non-negative constant.

***** ERROR *** CODE ADF2020:**

Unable to apply temperature-dependent-power-law material to the model since there is no temperature degree of freedom.

This message indicates that the temperature degree of freedom is inactive. If you intend to model the temperature, turn on the temperature degree of freedom together with all properly defined boundary conditions as well. If the temperature is not of interest, use another material model. Power-law material model could be the one that is closest to your original choice.

***** ERROR *** CODE ADF2021:**

Parameter A 1.19E-21 in the temperature-dependent power-law material model is invalid.

This message indicates that the value of parameter A is invalid in a temperature-dependent power-law material model. It is usually out of the range $[10^{-20}, 10^{20}]$. Check the definition of the material model for more details.

***** ERROR *** CODE ADF2022:**

Parameter N 1.19E-21 in the temperature-dependent power-law material model is invalid.

This message indicates that the value of parameter n is invalid in a temperature-dependent power-law material model. It is usually out of the range $[-10^{20},10^{20}]$. Check the definition of the material model for more details.

***** ERROR *** CODE ADF2023:**

Parameter C 1.19E-21 in the temperature-dependent power-law material model is invalid.

This message indicates that the value of parameter C is invalid in a temperature-dependent power-law material model. It is usually out of the range $[-10^{20},10^{20}]$. Check the definition of the material model for more details.

***** ERROR *** CODE ADF2024:**

Unable to apply the second-order material to the model since there is no temperature degree of freedom.

This message indicates that the temperature degree of freedom is inactive. If you intend to model the temperature, turn on the temperature degree of freedom together with all properly defined boundary conditions. If the temperature is not of interest, use another material model. The power-law material model could be the one that is closest to your original choice.

***** ERROR *** CODE ADF2025:**

Parameter A 1.19E-21 in the second-order material model is invalid.

This message indicates that the value of parameter A is invalid in a second-order material model. It is usually out of the range $[10^{-20},10^{20}]$. Check the definition of the material model for more details.

***** ERROR *** CODE ADF2026:**

Parameter C 1.19E-21 in the second-order material model is invalid.

This message indicates that the value of parameter C is invalid in a

second-order material model. It is usually out of the range $[-10^{20},10^{20}]$. Check the definition of the material model for more details.

***** ERROR *** CODE ADF2027:**

Parameter E 1.19E-21 in the second-order material model is invalid.

This message indicates that the value of parameter E is invalid in a second-order material model. It is usually out of the range $[-10^{20},10^{20}]$. Check the definition of the material model for more details.

***** ERROR *** CODE ADF2028:**

Parameter F 1.19E-21 in the second-order material model is invalid.

This message indicates that the value of parameter F is invalid in a second-order material model. It is usually out of the range $[-10^{20},10^{20}]$. Check the definition of the material model for more details.

***** ERROR *** CODE ADF2029:**

Parameter G 1.19E-21 in the second-order material model is invalid.

This message indicates that the value of parameter G is invalid in a second-order material model. It is usually out of the range $[-10^{20},10^{20}]$. Check the definition of the material model for more details.

***** ERROR *** CODE ADF2030:**

Parameter H 1.19E-21 in the second-order material model is invalid.

This message indicates that the value of parameter H is invalid in a second-order material model. It is usually out of the range $[-10^{20},10^{20}]$. Check the definition of the material model for more details.

***** ERROR *** CODE ADF2031:**

Unable to apply the turbulence material to the model since there is no turbulence degree of freedom.

This message indicates that the fluid model selected is not a two-equation turbulence model (K-ε or K-ω) while a turbulence material is used. Refer to the related chapters to see the available materials.

***** ERROR *** CODE ADF2032:**

Phase change requires temperature being presented. There is no temperature degree of freedom in this model.

This message indicates that the temperature degree of freedom is not active in this model. Whenever phase change is required or phase change boundary condition is applied, the temperature degree of freedom must be present.

***** ERROR *** CODE ADF2033:**

Parameter BetaT 1.19E-21 in the K-Omega material model is invalid.

This message indicates that the value of parameter β_θ is invalid in a turbulent K-ω material model. It is usually out of the range $[-10^{20}, 10^{20}]$. Check the definition of the material model for more details.

***** ERROR *** CODE ADF2034:**

Points are out of order in a curve. Program is unable to identify the type of the curve. Currently available status is:
"status_WHEN_ERROR_OCCURS".

This message indicates that a material curve is invalid because some points that define the curve are out of order. Tabulated material curves must be input in an increasing order with no repeated points. The status could be either “temperature-dependent material curves” or “time-dependent material curves”.

***** ERROR *** CODE ADF2035:**

Viscosity (XMU) 1.2345599E+20 is invalid.

This message indicates an invalid value of the fluid viscosity μ. It is usually out of the range $[10^{-20},10^{20}]$. Check the definition of the material model for more details.

***** ERROR *** CODE ADF2036:**

Density (RHO) 1.2345599E+20 is invalid.

This message indicates an invalid value of the fluid or solid density ρ. It is usually out of the range $[10^{-20},10^{20}]$. Check the definition of the material model for more details.

***** ERROR *** CODE ADF2037:**

Specific heat at constant pressure (Cp) 1.2345599E+20 is invalid.

This message indicates an invalid value of the specific heat at constant pressure C_p. It is usually out of the range $[10^{-20},10^{20}]$. Check the definition of the material model for more details.

***** ERROR *** CODE ADF2038:**

Specific heat at constant volume (Cv) 1.2345599E+20 is invalid.

This message indicates an invalid value of the specific heat at constant volume C_v. It is usually out of the range $[10^{-20},10^{20}]$. Check the definition of the material model for more details.

***** ERROR *** CODE ADF2039:**

Specific heat in solid element group (Cps) 1.2345599E+20 is invalid.

This message indicates an invalid value of the solid specific heat at constant pressure C_{p_s} that is required in a porous medium material. It is

usually out of the range $[10^{-20},10^{20}]$. Check the definition of the material model for more details.

***** ERROR/WARNING *** CODE ADF2040:**

Density in solid element group (ROHs) 1.2345599E+20 is invalid.

This message indicates an invalid value of the solid density ρ_s that is required in a porous medium material. It is usually out of the range $[10^{-20},10^{20}]$. Check the definition of the material model for more details.

***** ERROR *** CODE ADF2041:**

Heat conductivity (XKCON) 1.2345599E+20 is invalid.

This message indicates an invalid value of the heat conductivity k. It is usually out of the range $[10^{-20},10^{20}]$. Check the definition of the material model for more details.

***** ERROR *** CODE ADF2042:**

Porosity (PHI) 1.2345599E+20 is invalid.

This message indicates an invalid value of the porosity ϕ that is required in a porous medium material. It is usually out of the range $[0,1]$. Check the definition of the material model for more details.

***** ERROR *** CODE ADF2043:**

Permeability in x-direction (PERMx) 1.2345599E+20 is invalid.

This message indicates an invalid value of the x-permeability κ_1 that is required in a porous medium material. It is usually out of the range $[10^{-20},10^{20}]$. Check the definition of the material model for more details.

***** ERROR *** CODE ADF2044:**

Permeability in y-direction (PERMy) 1.2345599E+20 is invalid.

This message indicates an invalid value of the y-permeability κ_2 that is required in a porous medium material. It is usually out of the range $[10^{-20},10^{20}]$. Check the definition of the material model for more details.

***** ERROR *** CODE ADF2045:**

Permeability in z-direction (PERMz) 1.2345599E+20 is invalid.

This message indicates an invalid value of the z-permeability κ_3 that is required in a porous medium material. It is usually out of the range $[10^{-20},10^{20}]$. Check the definition of the material model for more details.

***** ERROR *** CODE ADF2046:**

Heat conductivity in solid element group (XKCONs) 1.2300E+20 is invalid.

This message indicates an invalid value of the solid heat conductivity k_s that is required in a porous medium material. It is usually out of the range $[10^{-20},10^{20}]$. Check the definition of the material model for more details.

***** ERROR *** CODE ADF2047:**

Parameter (A) 1.2345599E+20 in the power-law material model is invalid.

This message indicates that the value of parameter A is invalid in a power-law material model. It is usually out of the range $[10^{-20},10^{20}]$. Check the definition of the material model for more details.

***** ERROR *** CODE ADF2048:**

Parameter (N) 1.2345599E+20 in the power-law material model is invalid.

This message indicates that the value of parameter n is invalid in a

power-law material model. It is usually out of the range $[-10^{20}, 10^{20}]$. Check the definition of the material model for more details.

***** ERROR *** CODE ADF2049:**

The variable ID or/and equation ID have been replaced in the user-supplied-boundary-condition subroutine. An error may exist in the subroutine. The two IDs are stored in L(2) and L(3) respectively. Current values of them are 11 and 22.

This message indicates that an error has been found in the output of the user-supplied boundary condition subroutine. The two identification numbers, I_v and I_c stored at $L(2)$ and $L(3)$ respectively, are "read only" parameters in the user routine. However, their addresses have been found used after the function call. Check the program to make sure the passed-out parameters are saved at the correct locations in the passed-out arrays.

***** ERROR *** CODE ADF2050:**

Currently required element viscosity is not available. A wrong element may have been used. Only triangular/tetrahedron elements can be used if adaptive mesh option is applied.

This message indicates that possibly an unavailable element is used while the mesh repair option is selected. Currently, only triangular 3-node and tetrahedral 4-node elements are available if the mesh repair option is selected.

***** ERROR *** CODE ADF2051:**

Parameter (A) 1.2345599E+20 in the CARREAU material model is invalid.

This message indicates that the value of parameter A is invalid in a Carreau material model. It is usually out of the range $[10^{-20}, 10^{20}]$. Check the definition of the material model for more details.

***** ERROR *** CODE ADF2052:**

Parameter (N) 1.2345599E+20 in the CARREAU material model is invalid.

This message indicates that the value of parameter n is invalid in a Carreau material model. It is usually out of the range $[-10^{20},10^{20}]$. Check the definition of the material model for more details.

***** ERROR *** CODE ADF2053:**

X-gravitation (GRAVx) 1.2345599E+20 is invalid.

This message indicates an invalid value of the x-gravity g_1. It is usually out of the range $[-10^{20},10^{20}]$. Check the definition of the material model for more details.

***** ERROR *** CODE ADF2054:**

Y-gravitation (GRAVy) 1.2345599E+20 is invalid.

This message indicates an invalid value of the y-gravity g_2. It is usually out of the range $[-10^{20},10^{20}]$. Check the definition of the material model for more details.

***** ERROR *** CODE ADF2055:**

Z-gravitation (GRAVz) 1.2345599E+20 is invalid.

This message indicates an invalid value of the z-gravity g_3. It is usually out of the range $[-10^{20},10^{20}]$. Check the definition of the material model for more details.

***** ERROR *** CODE ADF2056:**

Surface tension (SIGMA) 1.2345599E+20 is invalid.

This message indicates an invalid value of the surface tension σ. It is usually out of the range $[10^{-20},10^{20}]$. Check the definition of the material model for more details.

***** ERROR *** CODE ADF2057:**

Reference temperature (Tc) 1.2345599E+20 is invalid.

This message indicates an invalid value of the reference temperature θ_0 in material data. It is usually out of the range $[-10^{20},10^{20}]$. Check the definition of the material model for more details.

***** ERROR *** CODE ADF2058:**

Rate of heat generated per unit volume (Qb) 1.2345599E+20 is invalid.

This message indicates an invalid value of the heat generation rate q^B in material data. It is usually out of the range $[-10^{20},10^{20}]$. Check the definition of the material model for more details.

***** ERROR *** CODE ADF2059:**

Coefficient of volume expansion (BETA) 1.2345599E+20 is invalid.

This message indicates an invalid value of the coefficient of volume expansion β in material data. It is usually out of the range $[-10^{20},10^{20}]$. Check the definition of the material model for more details.

***** ERROR/WARNING *** CODE ADF2060:**

Parameter (C1) 1.2345599E+20 in the turbulence model is invalid.

This message indicates an invalid value of the model constant c_1 in turbulent material data. It is usually out of the range $[10^{-20},10^{20}]$. Check the definition of the material model for more details.

***** ERROR/WARNING *** CODE ADF2061:**

Parameter (C2) 1.2345599E+20 in the turbulence model is invalid.

This message indicates an invalid value of the model constant c_2 in turbulent material data. It is usually out of the range $[10^{-20},10^{20}]$.

Check the definition of the material model for more details.

***** ERROR/WARNING *** CODE ADF2062:**

Parameter (C3) 1.2345599E+20 in the turbulence model is invalid.

This message indicates an invalid value of the model constant c_3 in turbulent material data. It is usually out of the range $[-10^{20}, 10^{20}]$. Check the definition of the material model for more details.

***** ERROR/WARNING *** CODE ADF2063:**

Parameter (CMU) 1.2345599E+20 in the turbulence model is invalid.

This message indicates an invalid value of the model constant c_μ in turbulent material data. It is usually out of the range $[10^{-20}, 10^{20}]$. Check the definition of the material model for more details.

***** ERROR/WARNING *** CODE ADF2064:**

Parameter (SIGMAK) 1.2345599E+20 in the turbulence model is invalid.

This message indicates an invalid value of the model constant σ_K in turbulent material data. It is usually out of the range $[10^{-20}, 10^{20}]$. Check the definition of the material model for more details.

***** ERROR *** CODE ADF2065:**

Parameter (SIGMAE) 1.2345599E+20 in the turbulence model is invalid.

This message indicates an invalid value of the model constant σ_ε in turbulent material data. It is usually out of the range $[10^{-20}, 10^{20}]$. Check the definition of the material model for more details.

***** ERROR/WARNING *** CODE ADF2066:**

Parameter (SIGMAT) 1.2345599E+20 in the turbulence model is invalid.

This message indicates an invalid value of the model constant σ_θ in turbulent material data. It is usually out of the range $[10^{-20},10^{20}]$. Check the definition of the material model for more details.

***** ERROR/WARNING *** CODE ADF2067:**

Von Kaman constant (VON) 1.2345599E+20 in the turbulence model is invalid.

This message indicates an invalid value of the Von Karman constant κ_0 in turbulent material data. It is usually out of the range $[10^{-20},10^{20}]$. Check the definition of the material model for more details.

***** ERROR/WARNING *** CODE ADF2068:**

Parameter (DW) 1.2345599E+20 in the turbulence model is invalid.

This message indicates an invalid value of the model constant d_w^+ in turbulent material data. It is usually out of the range $[10^{-20},10^{20}]$. Check the definition of the material model for more details.

***** ERROR *** CODE ADF2069:**

Bulk modulus of elasticity (KAPA) 1.23400E+20 is invalid.

This message indicates an invalid value of the bulk modulus κ used in a slightly compressible flow model. It is usually out of the range $[10^{-20},10^{20}]$. Check the definition of the material model for more details.

***** WARNING *** CODE ADF2070:**

Ratio of second viscosity (XMU2) 1.23400E+20 is invalid.

This message indicates a possibly invalid value of the rate of the second viscosity λ/μ in turbulent material data. It is usually out of the range $[-10,1]$. Check the definition of the material model for more details.

***** WARNING *** CODE ADF2071:**

Exponent for viscosity (SEXMU) 1.23400E+20 is invalid.

This message indicates a possibly invalid value of the exponent m_μ in the Sutherland's formula used for high-speed compressible flows. It is usually out of the range $[-10,10]$. Check the definition of the material model for more details.

***** WARNING *** CODE ADF2072:**

Parameter (S1XMU) -1.23400E+20 is invalid.

This message indicates a possibly invalid value of the constant temperature S_μ in Sutherland's formula used for high-speed compressible flows. It is usually out of the range $[-10^{-20},\infty)$. Check the definition of the material model for more details.

***** WARNING *** CODE ADF2073:**

Exponent for heat conductivity (SEXKC) 1.23400E+20 is invalid.

This message indicates a possibly invalid value of the exponent m_k in Sutherland's formula used for high-speed compressible flows. It is usually out of the range $[-10,10]$. Check the definition of the material model for more details.

***** WARNING *** CODE ADF2074:**

Parameter (S1XKC) -1.23400E+20 is invalid.

This message indicates a possibly invalid value of the constant temperature S_k in Sutherland's formula used for high-speed compressible flows. It is usually out of the range $[-10^{-20},\infty)$. Check the definition of the material model for more details.

***** ERROR *** CODE ADF2101:**

Length scale 1.23400E+20 in nondimensionalization is invalid.

This message indicates an invalid value of the length scale L^* in automatically nondimensional procedure. It is usually out of the range $[10^{-20},10^{+20}]$. Refer to the descriptions about nondimensional analyses.

***** ERROR *** CODE ADF2102:**

Velocity scale 1.23400E+20 in nondimensionalization is invalid.

This message indicates an invalid value of the velocity scale V^* in an automatically nondimensional procedure. It is usually out of the range $[10^{-20},10^{+20}]$. Refer to the descriptions about nondimensional analyses.

***** ERROR *** CODE ADF2103:**

Scale 1.2300E+20 for specific heats in nondimensionalization is invalid.

This message indicates an invalid value of the specific heat scale C^* in an automatically nondimensional procedure. It is usually out of the range $[10^{-20},10^{+20}]$. Refer to the descriptions about nondimensional analyses.

***** ERROR *** CODE ADF2104:**

Density scale 1.23400E+20 in nondimensionalization is invalid.

This message indicates an invalid value of the density scale D^* in an automatically nondimensional procedure. It is usually out of the range $[10^{-20},10^{+20}]$. Refer to the descriptions about nondimensional analyses.

***** ERROR *** CODE ADF2105:**

Temperature scale 1.23400E+20 in nondimensionalization is invalid.

This message indicates an invalid value of the temperature scale T^* in an automatically nondimensional procedure. It is usually out of the

range $[10^{-20},10^{+20}]$. Refer to the descriptions about nondimensional analyses.

***** ERROR *** CODE ADF2106:**

Mass-ratio scale 1.23400E+20 in nondimensionalization is invalid.

This message indicates an invalid value of the mass-ratio scale Φ^* in an automatically nondimensional procedure. It is usually out of the range $[10^{-20},10^{+20}]$. Refer to the descriptions about nondimensional analyses.

***** ERROR *** CODE ADF2107:**

X-coordinate datum 1.23400E+20 in nondimensionalization is invalid.

This message indicates an invalid value of the reference x-coordinate x_r^* in an automatically nondimensional procedure. It is usually out of the range $[-10^{12},10^{12}]$. Refer to the descriptions about nondimensional analyses.

***** ERROR *** CODE ADF2108:**

Y-coordinate datum 1.23400E+20 in nondimensionalization is invalid.

This message indicates an invalid value of the reference y-coordinate y_r^* in an automatically nondimensional procedure. It is usually out of the range $[-10^{12},10^{12}]$. Refer to the descriptions about nondimensional analyses.

***** ERROR *** CODE ADF2109:**

Z-coordinate datum 1.23400E+20 in nondimensionalization is invalid.

This message indicates an invalid value of the reference z-coordinate z_r^* in an automatically nondimensional procedure. It is usually out of the range $[-10^{12},10^{12}]$. Refer to the descriptions about

nondimensional analyses.

***** ERROR *** CODE ADF2110:**

Temperature datum 1.23400E+20 in nondimensionalization is invalid.

This message indicates an invalid value of the reference temperature θ_r^* in an automatically nondimensional procedure. It is usually out of the range $[-10^{12}, 10^{12}]$. Refer to the descriptions about nondimensional analyses.

***** ERROR *** CODE ADF2111:**

Control flag (IFGNDM) -1 in nondimensionalization is invalid.

This message indicates an invalid control flag in an automatically nondimensional procedure. A wrong data file may cause this error.

Whenever a possibly wrong data file is encountered, make sure the following requirements are satisfied.

- A correct data file is used.
- The versions of AUI and ADINA-F are consistent. It is suggested that always use the latest consistent versions.
- If the data file was generated on a different machine where ADINA-F is installed, the data file must be portable. In general, use a formatted (ASCII) data file. Usually, binary data files are compatible between Unix machines. They are not compatible with Windows PCs and supercomputers.
- If you have modified the *.dat file, make sure that the format is correct. It is not recommended (although not restricted) to edit parameters in the data file.

***** ERROR *** CODE ADF2112:**

Analysis flag (IFGNDI) -1 in nondimensionalization is invalid.

This message indicates an invalid control flag in an automatically nondimensional procedure. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2113:**

Output flag (IFGNDO) -1 in nondimensionalization is invalid.

This message indicates an invalid control flag in an automatically nondimensional procedure. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2151:**

Starting time (TSTART) 1.23400E+20 is invalid.

This message indicates an invalid value of the restart time. It is usually out of the range $[-10^{-20}, \infty)$. Check the latest time that the solution is converged in the previous run. The restart time specified in the restart run must be the same as that saved in the *.res file. Refer to the descriptions about restart analyses.

***** ERROR *** CODE ADF2152:**

Number of nodes (NNODE) -1 is invalid.

This message indicates an invalid number that represents the number of nodes. It is usually a non-positive number. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2153:**

Execution flag (MODEX) 1234 is invalid.

This message indicates an invalid control flag. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2154:**

Number of element groups (NELMG) -1 is invalid.

This message indicates an invalid number that represents the number of element groups. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2155:**

Number of time step blocks (NDTBK) -1 is invalid.

This message indicates an invalid number that represents the number of time step blocks. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2156:**

Number of skew systems (NSKEW) -1 is invalid.

This message indicates an invalid number that represents the number of skew systems defined in this model. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2157:**

Rotational flag (IROTA) 1234 is invalid.

This message indicates an invalid flag for centrifugal load data. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2158:**

Flag (ICOMP) 1234 for fluid flow type is invalid.

This message indicates an invalid flag that represents the flow model selected. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2159:**

Turbulence model type (IFGTUR) 1234 is invalid.

This message indicates an invalid flag that represents the selected turbulence model type. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2160:**

Flag (IHYDRO) 1234 for including hydrostatic pressure is invalid.

This message indicates an invalid flag that represents the treatment of the hydrostatic pressure. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2161:**

Flag (METHOD) 1234 for equilibrium method is invalid.

This message indicates an invalid flag that represents the selected equilibrium iteration method. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2162:**

Maximum number of equilibrium iterations (ITEQLM) -1 is invalid.

This message indicates an invalid number that represents the specified maximum number of iterations that are permitted in equilibrium iterations. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2163:**

Number of functions (NFN) -1 is invalid.

This message indicates an invalid number that represents the number of one-dimensional functions defined in this model. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2164:**

Maximum number of function points (NFP) -1 is invalid.

This message indicates an invalid number that represents the maximum number of points defined in one-dimensional functions in this model. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2209:**

Pressure points are too close in material data. Distance = 2.19E-21
Check the input related to material data.

This message indicates that a pressure-dependent material curve is invalid because some pressure points that define the curve are out of order. Tabulated material curves must be input in an increasing order and no repeated points.

***** ERROR *** CODE ADF2210:**

Temperature points are too close in material data. Distance = 2.19E-21
Check the input related to material data.

This message indicates that a temperature material curve is invalid because some temperature points that define the curve are out of order. Tabulated material curves must be input in an increasing order and no repeated points.

***** ERROR *** CODE ADF2217:**

78 errors have been found in NPAR parameters in element group data.

This message indicates that a few errors have been identified in the element group control data. Check the earlier printed error/warning messages. In case a wrong data file is used, also refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2218:**

Element birth time is larger than element death time:
element (label) = 1;
element group number = 2;
birth time = 3.0009E-02;
death time = 2.0020E-02.

This message reports an error found in the element group data. An element birth time has been found larger than the element death time. It is required that birth time is always smaller than the death time. It is also possible that a wrong data file is used. In this case, also refer to the

suggestions for message ADF2111.

***** ERROR *** CODE ADF2220:**

Element number -1 in element group 2 is invalid.
Data file could be wrong.

This message reports an error found in the element group data. A non-positive element label has been encountered. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2221:**

Nodal label 1 in element connectivity data is invalid:
element (label) = 1;
element group number = 2.
Data file could be wrong.

This message reports an error found in the element group data. A node that forms an element cannot be recognized. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2222:**

Inconsistency in element group control data (NPAR(6)) and element connectivity data. In element 890, element group 213,
the node 11 is defined in a skew system, which was not shown
in control data.

This message reports inconsistent information shown in the element group data and in global control data. The information is related to the skew systems defined in this model. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2223:**

Material data set number -1 is invalid in element 890,
element group 112.
Data file could be wrong.

This message reports an error found in the element group data. An element group can have only one fluid material data set. However, the

number shown here is inconsistent with the assumption. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** WARNING *** CODE ADF2224**:

Model may be unstable because temperature or pressure is too small (1.19E-21). Local temperature or pressure has been adjusted by applying correction factor 1.230E-01 to node 890.

This message reports a possible error in the model. Whenever a compressible flow model (low-speed or high-speed) is used, the pressure and temperature must not be zero anywhere in the model. Refer to the formulations for compressible flow models for more details on the assumptions and limitations.

Recall that the state equation used in compressible flows is $p = (C_p - C_v)\rho\theta$. The pressure must be absolute pressure and the temperature must be absolute temperature. The state equation must be satisfied through all the input data, including materials, loads, boundary conditions, initial conditions, etc.

If this message is encountered at the first iteration in the first time step, usually some wrong data has been input. You may perform a test by imposing conditions such that the solution is expected to be very close to the initial conditions. In the same test model, specify a large value for equilibrium iteration tolerance so the solution will be saved after the first iteration. With the help of the printed information in the *.out file and the help of visualization of the saved solution, find the locations where the solutions are unreasonable.

If the pressure and the temperature are not too small, an improper unit system is also possible. In this case, either use another proper system of units or use the automatically nondimensional procedure by specifying proper scales. Refer to the related topics for more details.

***** ERROR *** CODE ADF2226:**

Incorrect initial conditions have been found at node 890. This may occur if pressure and/or temperature are too small or Mach number is too large.
Check the input related to initial conditions and material data.

Currently available information is:

pressure = 1.23450E-21;
x-velocity = 2.23449E-21;
y-velocity = 3.23449E-21;
z-velocity = 4.23449E-21;
temperature = 5.23449E-21;
Cp = 6.23449E-21;
Cv = 7.23449E-21.

This message reports a possible error in the initial condition. Possibly a zero pressure or a zero temperature has been specified in compressible flow models. Refer to the suggestions for message ADF2224.

***** ERROR *** CODE ADF2240:**

Incorrect constraint equation data: Constraint equation 890 is at position 11112.
Data file could be wrong.

This message reports an error found in the constraint equation data. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2241:**

Incorrect constraint equation data: too many master nodes 2222 in constraint equation 890. The largest possible number is 2333.
Data file could be wrong.

This message reports an error found in the constraint equation data. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2242:**

Inconsistency between constraint equation data and nodal data: the slave node 11112 defined in the constraint equation 890 is unable to be identified as a slave node based on nodal information.
Data file could be wrong.

This message reports an error found in the constraint equation data. A wrong data file may cause this error. Refer to the suggestions for

message ADF2111.

***** ERROR *** CODE ADF2243:**

Inconsistency between constraint equation data and nodal data: a master node defined in constraint equation 890 has no degree of freedom.

This message reports an error found in the constraint equation data. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2244:**

A node defined in constraint equation 890 is out of label range.
Data file could be wrong.

This message reports an error found in the constraint equation data. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2245:**

A variable identification direction in constraint equation 890 is out of label range.
Data file could be wrong.

This message reports an error found in the constraint equation data. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

*** ERROR *** CODE ADF2246:

Variable identification directions in master nodes and slave nodes are not identical in constraint equation 890.
Data file could be wrong.

This message reports an error found in the constraint equation data. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2247:**

```
In high speed compressible flows, all degrees of freedom of a slave node
must be constrained to the same degrees of freedom of its master node.
Error has been found in the following data set:
  constraint equation             =       11;
  slave nodal label               =       22;
  variable identification direction =     33.
Check the input related to constraint equations.
```

This message reports an error found in the constraint equation data. In high-speed compressible flows, because of the highly nonlinear property of the solution variables, ADINA-F does not accept that a solution variable is constrained to a different solution variable. It is also improper if only a part of solution variable is constrained. A wrong data file may cause this error as well. In this case, refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2248:**

```
Incorrect combination of variables "S_V" in boundary condition 119.
```

This message reports an error found in a special boundary condition in the high-speed compressible flow model. The combination of the variables is not available. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR/WARNING *** CODE ADF2249:**

```
Unreasonable solutions obtained at boundary node 1118 (index) in
boundary condition 19 where the combination of variables is "S_V".
Values of boundary variables may be modified in case it is necessary.

The values of inside primitive variables are:

  p =  1.1234559E+11
  u =  1.1234559E+11
  v =  1.1234559E+11
  w =  1.1234559E+11
  t =  1.1234559E+11

The computed values of outside primitive variables are:

  p =  1.1234559E+11
  u =  1.1234559E+11
  v =  1.1234559E+11
```

w = 1.1234559E+11
t = 1.1234559E+11

Current values of Cp and Cv are:

Cp = 3.1277770E+11
Cv = 3.2277771E+11

Current boundary variables are:

3.32769E+11, 3.42769E+11, 3.52769E+11.

The above values may be in dimensionless form if automatic nondimensionalization is used.

This message reports an unreasonable solution obtained in a special boundary condition in the high-speed compressible flow model.

When this message is encountered at the first iteration in the first time step, make sure the following conditions are satisfied:

- The solution variables specified in boundary conditions satisfy the state equation. In particular, the resultant Mach number is correct.
- The initial conditions are correctly specified. In particular, no zero pressure or temperature are specified.
- The material data are correctly input, particularly the C_p and C_v.

When this message is encountered in later iterations or in later time steps, the solution procedure is probably divergent. Check the latest converged solution to find any possible errors. If no solutions are saved yet, use a large equilibrium tolerance to save the solution after the first iteration. Refer to the suggestions described under the topic "model preparation and testing" for more details.

***** WARNING *** CODE ADF2250:**

Pivot has been assigned to a boundary equation, where the boundary variable combination is "S_V" in boundary condition (index) 123456.

This message reports an ill conditioned matrix in a special boundary condition in the high-speed compressible flow model. The possible causes of this error are likely the same as those described in message ADF2249.

***** ERROR *** CODE ADF2251:**

Unidentified flow regime 1234 in boundary condition 222222.
Data file could be wrong.

This message reports an error found in a special boundary condition in the high-speed compressible flow model. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2252:**

Unidentified variable ID 111 in boundary condition 1119.
Data file could be wrong.

This message reports an error found in a special boundary condition in the high-speed compressible flow model. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2253:**

Incorrect combination of variables in boundary condition 11. Their identification flags are 22, 33 and 44.

This message reports an error found in a special boundary condition in the high-speed compressible flow model. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2254:**

Failed in solving boundary equations in boundary condition 11.

Possible causes are:

* Incorrect boundary conditions have been applied;
* Physical model is unstable;
* Solution procedure is divergent.

This message reports a singularity found in a special boundary condition in the high-speed compressible flow model. The possible causes of this error are likely the same as those described in message ADF2249.

***** ERROR *** CODE ADF2255:**

The boundary equation is singular in boundary condition 11.

Possible causes are:

* Incorrect boundary conditions have been applied;
* Physical model is unstable;
* Solution procedure is divergent.

This message reports a singularity found in a special boundary condition in the high-speed compressible flow model. The possible causes of this error are likely the same as those described in message ADF2249.

***** ERROR *** CODE ADF2256:**

Failed in solving the fluid state equation.

Possible causes are:

* Incorrect boundary conditions have been applied;
* Physical model is unstable;
* Solution procedure is divergent.

Currently available information is:

D = 1.120E+11;
DI = 1.219E+11;
P = 1.319E+11;
T = 1.419E+11;
CP = 1.520E+11;
CPP = 2.119E+11;
CPT = 2.220E+11;
CV = 2.319E+11;
CVP = 2.420E+11;
CVT = 2.519E+11.

The above values may be in dimensionless form if automatic nondimensionalization is used.

This message reports a singularity found in a special boundary condition in the high-speed compressible flow model. The possible causes of this error are likely the same as those described in message ADF2249.

***** ERROR *** CODE ADF2257:**

Distributed loads (normal-traction, heat-flux, etc.) cannot be applied to internal faces. Check the load applied to the face that is formed by the Nodes: (1,8,3,78).

This message reports a load that has been applied to an internal face. Distributed loads, such as normal-tractions, heat-fluxes, etc., should never be applied to internal faces. This could be a input error in the model.

***** ERROR *** CODE ADF2305:**

The turbulence K-Omega model is currently not available for high speed compressible flows. Turbulence K-Epsilon is available.

This message indicates that the turbulent K-ω is currently not available in the high-speed compressible flow model. Only the turbulent K-ε can be used for the high-speed compressible flows. You may change the turbulence model type and rerun the model.

***** ERROR *** CODE ADF2306:**

Force is not available at this stage.
This could be an internal error or the current working environment has been accidentally broken.

This message indicates that the program is unable to compute the fluid force or stress because of an unknown reason. One possible reason is that some temporary files have been removed (not by the program). Another possibility is that a wrong data file is used. In this case, refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2307:**

The matrix size is too large. Current 32-bit program versions can only

handle problems with less than 2147483647 matrix elements.

This message indicates that the size of the matrix surpasses the limit 2^{+31} on a 32-bit machine. If the solver selected is the Gauss elimination method (SKYLINE), choose the sparse or iterative solvers instead. The iterative solver RPBCG uses the least memory.

***** ERROR *** CODE ADF2308:**

Number of points 1119 in boundary geometry 56 is invalid.
Data file could be wrong.

This message reports an error found in the boundary geometry data. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2309:**

Number of cells 1119 in boundary geometry 56 is invalid.
Data file could be wrong.

This message reports an error found in the boundary geometry data. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2404:**

Time value in temperature file is out of order.
The file may not be generated by ADINA program or a wrong temperature file was used.

This message reports an error found in the temperature file. Do you really want to apply the temperature saved in the temperature file? Probably not but have accidentally selected that choice. However, if you want, the temperature file must be generated by ADINA program. The temperature solutions are saved on nodes in every time step. The time range saved must cover the time range in the current model. The nodes saved must also be the same as those in the current model.

A wrong data file may cause this error as well. In this case, refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2701:**

Fluid node (label) 99999999 is not attached to solid fluid-structure interface 22. The relative distance to solid boundary element 33 is 9.99E-22.
The fluid nodal coordinates are (1.00E+01, 2.00E+01, 3.00E+01).
The points and their coordinates of the solid boundary element are:
point 444444 = (1.400E+01, 2.400E+01, 3.400E+01)
point 555555 = (1.500E+01, 2.500E+01, 3.500E+01)
point 666666 = (1.600E+01, 2.600E+01, 3.600E+01)

Possible causes are:

* Fluid-structure interfaces in fluid and solid models are not coincident;
* Solid boundary elements are too coarse compared with fluid boundary elements on the same fluid-structure interface.

Recall that fluid-structure interfaces are specified in both fluid and solid models. They must be coincident with each other. However, it is normal to have a small distance between fluid nodes and solid fluid-structure interfaces if meshes of interfaces in both models are not matched.

The relative distance here is defined as the ratio of the distance between the fluid node and the closest solid boundary element to the length of the solid boundary element.

This message reports a distance, in three dimensions, that exists between two fluid-structure interfaces that are defined in the fluid model and the solid model respectively. Recall that the two interfaces specified in the fluid and the solid model form a pair to define the interaction; they represent the same geometric surface.

Since the two models are allowed to be generated separately, the meshes on the interfaces are likely to be unmatched. Reasonably small gaps are therefore expected. However, when a relative distance is too large, the program treat it as an error. This situation may occur if two discrepant geometries are used in fluid and solid models. It may also occur if the sizes of the two meshes are too deviated, particularly along boundary locations of sharp curvatures. Excessively fine details on the boundary geometry may also cause this problem if the fluid and solid nodes are unmatched. Refer to the related chapters for more details.

***** ERROR *** CODE ADF2702:**

Fluid node (label) 99999999 is not attached to solid fluid-structure interface 22. The relative distance to solid boundary element 33 is 9.99E-22.
The fluid nodal coordinates are (1.00E+01, 2.00E+01, 3.00E+01)
The points and their coordinates of the solid boundary element are:
point 444444 = (1.400E+01, 2.400E+01, 3.400E+01)
point 555555 = (1.500E+01, 2.500E+01, 3.500E+01)

Possible causes are:

* Fluid-structure interfaces in fluid and solid models are not coincident;
* Solid boundary elements are too coarse compared with fluid boundary elements on the same fluid-structure interface.

Recall that fluid-structure interfaces are specified in both fluid and solid models. They must be coincident with each other. However, it is normal to have a small distance between fluid nodes and solid fluid-structure interfaces if meshes of interfaces in both models are not matched.

The relative distance here is defined as the ratio of the distance between the fluid node and the closest solid boundary element to the length of the solid boundary element.

This message reports a distance, in two dimensions, that exists between two fluid-structure interfaces that are defined in the fluid model and the solid model respectively. Recall that the two interfaces specified in the fluid and the solid model form a pair to define the interaction; they represent the same geometric surface.

This problem is the same as described for message ADF2701.

***** ERROR *** CODE ADF2703:**

Fluid node (label) 999999 is not attached to boundary geometry 22.
The relative distance to boundary cell 33 is 0.10E-20.
The fluid nodal coordinates are (1.00E+01, 2.00E+01, 3.00E+01)
The points and their coordinates of the boundary cell are:
point 444444 = (1.400E+01, 2.400E+01, 3.400E+01)
point 555555 = (1.500E+01, 2.500E+01, 3.500E+01)
point 666666 = (1.600E+01, 2.600E+01, 3.600E+01)

It is normal to have a small distance between fluid nodes and boundary geometry. The relative distance here is defined as the ratio of the distance between the fluid node and the closest boundary cell to the length of the cell.

Note: The boundary geometry is internally generated by the program.

This message reports an error found in the three-dimensional boundary geometry data. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2704:**

Fluid node (label) 999999 is not attached to boundary geometry 22.
The relative distance to boundary cell 33 is 9.99E-22.
The fluid nodal coordinates are (1.00E+01, 2.00E+01, 3.00E+01)
The points and their coordinates of the boundary cell are:
point 444444 = (1.400E+01, 2.400E+01, 3.400E+01)
point 555555 = (1.500E+01, 2.500E+01, 3.500E+01)

It is normal to have a small distance between fluid nodes and boundary geometry. The relative distance here is defined as the ratio of the distance between the fluid node and the closest boundary cell to the length of the cell.

Note: The boundary geometry is internally generated by the program.

This message reports an error found in the two-dimensional boundary geometry data. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2705:**

Dimension 111 in boundary geometry 56 is invalid.
Data file could be wrong.

This message reports an error found in the boundary geometry data. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2706:**

Identification type 111 in boundary geometry 56 is invalid.
Data file could be wrong.

This message reports an error found in the boundary geometry data. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2707:**

Time function number 111 in boundary geometry 56 is invalid.
time functions are applied.

This message reports an error found in the boundary geometry data. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2708:**

Birth time -1.0E-12 or/and death time -2.1E-12 in boundary geometry 56 are invalid.
Data file could be wrong.

This message reports an error found in the boundary geometry data. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2710:**

Neighbor cells of point 13456 are not connected properly in boundary geometry 156. 22 neighbor cells are found, but only 11 of them are connected.
This could be an internal error or a wrong data file was used.

This message reports an error found in the boundary geometry data. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2711:**

Point number 55 in boundary geometry 22 is out of range [1,11]. Current point index is 44 in cell 33.
This could be an internal error or a wrong data file was used.

This message reports an error found in the boundary geometry data. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2712:**

```
Incorrect data was found in boundary condition 21: invalid combination
of the following parameters:
  identification type     =  21;
  moving type             =  31;
  geometry type           =  41;
  boundary geometry (label) =  51.
This could be an internal error or a wrong data file was used.
```

This message reports an error found in boundary condition data. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2713:**

```
Cell 22 in boundary geometry 11 has zero length/area.
This could be an internal error or a wrong data file was used.
```

This message reports an error found in the boundary geometry data. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2726:**

```
Fluid-structure interface (label) 22 in boundary condition 11 is
invalid. This could be an internal error or wrong data file was used.
```

This message reports an error found in a boundary condition data. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2727:**

```
Boundary geometry (label) 22 in boundary condition 11 is invalid.
This could be an internal error or a wrong data file was used.
```

This message reports an error found in boundary condition data. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2728:**

```
Boundary condition (label) 11 has been applied to an improper location
```

```
   The left    side of the location is fluid element group
   The right side of the location is fluid element group
A boundary condition can only be applied to certain locations where they
are physically meaningful.
```

This message reports an error found in a boundary condition data. This condition has been applied to a location that has no physical meaning. For example, a wall condition cannot be applied to lines inside a 2D fluid domain.

***** ERROR *** CODE ADF2735:**

```
No conditions for moving mesh are found in boundary condition 5.
This could be an internal error or a wrong data file was used.
```

This message reports an error found in boundary condition data. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2737:**

```
Parameter SLIPC (=55) in boundary condition 11 is invalid.
Check the input related to this condition.
```

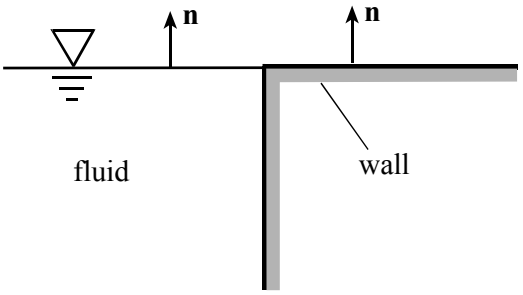
This message reports an error found in boundary condition data. The *slipc* parameter indicates the type of velocity condition in a defined direction. You may check the input of that boundary condition. A wrong data file may also cause this error. In this case, refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2738:**

```
BCD_TYPE (slave) and the attached solid boundaries (master) are
parallel. Current available information is
   slave condition label  =       1111;
   slave nodal label      =       2222;
   slave normal           = 1.000E+03, 2.000E+03, 3.000E+03
   master condition label =       3333;
   master normal          = 4.000E+03, 5.000E+03, 6.000E+03
```

This message reports an error found while treating a boundary condition data. The boundary condition could a free-surface condition, a fluid-fluid condition or a phase-change condition. When a free

surface, for example, touches a wall, the wall functions as a frontier. However, program finds out the normal direction of the free surface is coincident with the normal direction of the wall and, therefore, unable to continue the computation.



A wrong data file may cause this error as well. In this case, refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2750:**

Moving type 22 in boundary condition 11 is invalid.
This could be an internal error or a wrong data file was used.

This message reports an error found in boundary condition data. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2751:**

Type 22 in boundary condition 11 is invalid.
This could be an internal error or a wrong data file was used.

This message reports an error found in boundary condition data. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2752:**

The parameter 22 that indicates number of real numbers in boundary condition 11 is invalid.
This could be an internal error or a wrong data file was used.

This message reports an error found in boundary condition data. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2753:**

Number 22 for the integers in boundary condition 11 is invalid.
This could be an internal error or a wrong data file was used.

This message reports an error found in boundary condition data. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2754:**

Number of boundary nodes 22 in boundary condition 11 is invalid.
This could be an internal error or a wrong data file was used.

This message reports an error found in boundary condition data. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2755:**

Boundary node (label) 99999 is unable to be connected to other boundary nodes in boundary condition 11.

Possible causes are:

* While a boundary condition is applied to nodes (e.g., defined in PATRAN, I-DEAS, etc.) incorrect or incomplete boundary nodes may have been picked.
* An internal line/surface is applied with a condition that should only applied to boundary.
* A wrong data file was used.

This message reports an error found in boundary condition data. Boundary nodes in the same boundary condition must be able to be connected to each other and form boundary elements that represent discretized boundary lines (in two dimensions) or surfaces (in three dimensions).

If the model is generated using other programs, the conditions are

usually applied to nodes rather than on geometry lines and surfaces. When there is data missed while they are imported to ADINA system (usually through the TRANSOR program), the boundary nodes may not be connectable. Sometimes, these programs allow conditions only applied to nodes, you may pick incomplete nodes along a boundary. This may also cause this problem.

The boundary condition of the reported type here can only be applied to boundaries. Therefore, the boundary elements are facing fluid elements on only one side of them. If a boundary condition is applied to internal lines or surfaces, where both sides of them facing fluid elements, the boundary elements are unable to be identified and generated. In this case, the boundary condition is applied to a wrong location.

A wrong data file may cause this error as well. In this case, refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2756**:

Variable identification number 22 is out of range [1,3] in
user-supplied boundary condition 11.
This could be an internal error or a wrong data file was used.

This message reports an error found in a user-supplied boundary condition data. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2757:**

Follower boundary node 33 is defined at a wrong location. It should only
appear in moving boundary condition where the geometry is line.
Current boundary geometry type is 22 in boundary condition 11.
This could be an internal error or a wrong data file was used.

This message reports an error found in boundary condition data. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** WARNING *** CODE ADF2759:**

Geometry label 33 in boundary condition 11 is invalid. The
condition type is 22. In conditions of this type, no geometries are

expected. This could be an internal error or a wrong data file was used.

This message reports a potential error encountered in boundary condition data. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2760:**

Geometry type 33 in boundary condition 11 is invalid.
Current boundary condition type is 22. Geometry type should be 2 or 3 indicating a line or a surface respectively.
This could be an internal error or a wrong data file was used.

This message reports an error found in boundary condition data. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2761:**

Follower nodes in boundary condition 11 are impossible to follow leader nodes that are defined in boundary condition 22.
This could be an internal error or a wrong data file was used.

This message reports an error found in boundary condition data. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2762:**

Inconsistent boundary dimensions 22 and 33 are generated. Current boundary condition is 11.
This could be an internal error or a wrong data file was used.

This message reports an error found in boundary condition data. In general, a boundary condition (particularly a special boundary condition) should be applied to lines (in two-dimensional models) or surfaces (in three-dimensional models). It is not allowed that a boundary condition is applied to lines and surfaces in the same model.

A wrong data file may also cause this error. In this case, refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2764:**

Boundary node (label) 999999 defined in boundary condition 1111
is out of possible range.
This could be an internal error or a wrong data file was used.

This message reports an error found in boundary condition data. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2765:**

Follower node (label) 99999999 in boundary condition 11 is invalid.
This could be an internal error or a wrong data file was used.

This message reports an error found in boundary condition data. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2766:**

Leader node (label) 99999999 in boundary condition 11 is invalid.
This could be an internal error or a wrong data file was used.

This message reports an error found in boundary condition data. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2767:**

Leader nodal index 22 in boundary condition 11 is invalid.
This could be an internal error or a wrong data file was used.

This message reports an error found in boundary condition data. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** WARNING *** CODE ADF2771:**

Open value 1.1200E+00 is smaller than close value 2.2200E+00 in gap boundary condition 11. This may result in unreasonable oscillations in open/close status in this gap.

As a general guide, always let open value be slightly bigger than close

value.

This message reports a potential error found in a gap boundary condition data. The open value and close value in a gap condition controls the open/close status of the gap. If the open value is smaller than the close value, the gap may alternatively open and close in each time steps. Usually, this is unwanted physical situation. As a general rule, the open value must be slightly larger than the close value. Refer to the description of the gap condition.

***** WARNING *** CODE ADF2772:**

Close value 1.1230E-13 is too small in gap boundary condition 22.
It has been changed to 2.2221E+00.

This message reports a potential error found in a gap boundary condition data. When the close value is smaller than 10^{-12}, this warning message is printed and the close value is modified to 10^{-12}. When this value is too small, the meshes near the gap may be overlapped as the gap approaches to the closed position.

***** ERROR *** CODE ADF2773:**

Type 22 in gap boundary condition 11 is invalid.
This could be an internal error or a wrong data file was used.

This message reports an error found in boundary condition data. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2775:**

Initial open/close status 22 in gap boundary condition 11 is invalid.
This could be an internal error or a wrong data file was used.

This message reports an error found in a gap boundary condition data. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2776:**

Inconsistent dimensions of boundary elements in boundary condition 11.
Each two-dimensional boundary element is expected to be connected by two nodes.
Current boundary element is connected by three boundary nodes (their indices are 22, 33 and 44), corresponding to nodes (labels)
55, 66 and 77.
This could be an internal error or a wrong data file was used.

This message reports an error found in boundary condition data. The cause of this error is essentially the same as described for message ADF2762.

***** ERROR *** CODE ADF2777:**

Inconsistent dimensions of boundary elements in boundary condition 11.
Each three-dimensional boundary element is expected to be connected by three nodes.
Current boundary element is connected by two boundary nodes (their indices are 22 and 33), corresponding to nodes (labels) 55
and 66.
This could be an internal error or a wrong data file was used.

This message reports an error found in boundary condition data. The cause of this error is essentially the same as described for message ADF2762.

***** ERROR *** CODE ADF2783:**

Inconsistent boundary conditions have been found. Any two nodes on the same side of an element (node=9 or 27) must be assigned same boundary conditions. Currently available information is:
boundary condition = 11;
element group number = 22;
element number = 33;
first node on the side = 44;
second node on the side = 55.

If boundary conditions are applied to geometries, this error should never occur unless a wrong data file was used. However, if they are applied to nodes (e.g., defined in PATRAN, I-DEAS, etc.), incorrect or incomplete boundary nodes may have been picked.

This message reports an error found in boundary condition data. The cause of this error is essentially the same as described for message ADF2762.

***** ERROR *** CODE ADF2801:**

Memory is too little for the computation to continue. The block length 1119 is smaller than the number of equations.

This message indicates that memory assigned to ADINA-F is far less than required. Refer to the explanations under the topic "Use of memory and disk" for more details.

A wrong data file may also cause this error. In this case, refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF2803:**

Memory is too little for the computation to continue. Too many blocks (123456) are generated.

This message indicates that memory assigned to ADINA-F is far less than required. Refer to the explanations under the topic "Use of memory and disk" for more details.

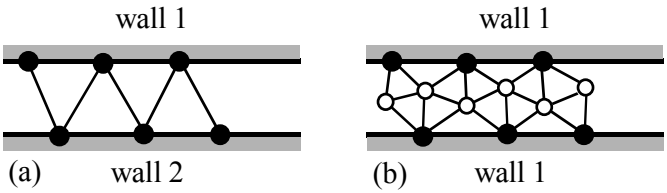
A wrong data file may also cause this error. In this case, refer to the suggestions for message ADF2111.

***** WARNING *** CODE ADF2804:**

Boundary side is connected to two elements in boundary condition 11.
Nodes that form the side are 22 and 33.
This side will not be treated as boundary element.

This message reports a possible error in the boundary condition data. If the elements are generated using a free mesh generator, it is possible there are some elements in which each corner nodes have been assigned the same boundary condition. If the elements are generated using a rule based mesh generator, this situation rarely occurs although still possible.

However, certain incorrect applications of boundary condition are also possible. In the example shown in the next figure (a), if the same wall condition is applied to both top and bottom lines, the program will be confused. To solve the problem, you may either apply two wall conditions of different labels to the top and to the bottom respectively as shown in (a) or use more than one layer elements in the fluid domain as shown in (b).



If the model is generated using other programs, the conditions are usually applied to nodes rather than on geometry lines and surfaces. When some internal nodes are picked while a boundary condition is applied, this error may also occur.

***** WARNING *** CODE ADF2805:**

Boundary face is connected to two elements in boundary condition 11.
Nodes that form the side are 22, 33 and 44.
This face will not be treated as boundary element.

This message reports a possible error in the boundary condition data. This is the three-dimensional case of the message ADF2804.

***** WARNING *** CODE ADF2806:**

Boundary side is connected to two elements in boundary condition 11.
The center node on the side is 22.
This side will not be treated as boundary element.

This message reports a possible error in the boundary condition data. The cause of it is possibly the same as described for message ADF2804. However, this message indicates that element is either quadrilateral 9-node or brick 27-node. In these elements, this message usually points to an error in the element. It is possible that a boundary condition that can only be applied to boundaries has been applied to an internal line or surface.

***** ERROR *** CODE ADF3000:**

No enough memory to continue.

This message indicates that the memory assigned to ADINA-F is far

less than required. Refer to the explanations under the topic "Use of memory and disk" for more details.

A wrong data file may also cause this error. In this case, refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF3001:**

```
No enough memory to continue. Currently available information about the
required array is:
  type          (LW      ) =        11;
  dimension     (LD      ) =        22;
  device unit   (IUN     ) =        33;
  short of words (IAM-IA2-1) =       44;
  identification string     = ID_STRING.
```

This message indicates that the memory assigned to ADINA-F is not enough. Refer to the explanations under the topic "Use of memory and disk" for more details.

A wrong data file may also cause this error. In this case, refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF3003:**

```
No pressure degree of freedom in the model.
Check the flow assumptions or the definitions in "MASTER" command.
```

This message indicates that there is no pressure degree of freedom in this model. Whenever fluid velocity is considered, the pressure must be included in the model.

A wrong data file may also cause this error. In this case, refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF3004:**

```
There are no active degrees of freedom.
Check the flow assumptions or the definitions in "MASTER" command.
```

This message indicates that there are no solution variables included in this model. It is possible that you may have created or modified your model in commands. In MASTER command, you probably have turned off all degrees of freedom.

A wrong data file may also cause this error. In this case, refer to the

suggestions for message ADF2111.

***** ERROR *** CODE ADF3005:**

Variable identification flag 11 (IDOF) is invalid.
Check the flow assumptions or the definitions in "MASTER" command.
It is also possible that a wrong data file has been used.

This message reports an error that is related to the overall solution degrees of freedom. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF3006:**

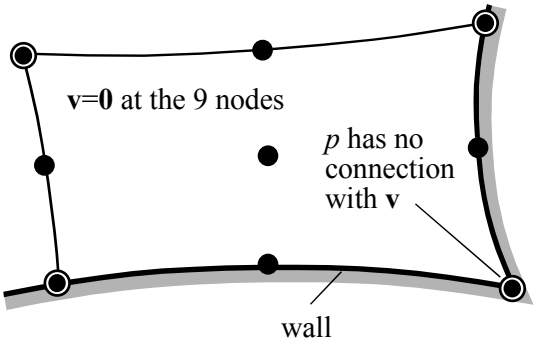
Flag for temperature file is invalid.
Possibly a wrong data file was used.

This message reports an error that is related to the usage of the temperature file. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** WARNING *** CODE ADF3007:**

Pressure equation 111111 at node 222222 is unable to be reasonably connected to its neighbor velocity degrees of freedom. System matrix could be singular or the boundary condition near the node could be incorrect.

This message reports a potential error in this model. The solution variable pressure must always be coupled with the solution variable velocities. However, program has identified a pressure that has no connections with velocities. Possibly the velocities around that pressure node have been wrongly fixed. This error can only occur when quadrilateral 9-node and brick 27-node elements are used. A typical example is shown in the figure below.



A wrong data file may also cause this error. In this case, refer to the suggestions for message ADF2111.

***** WARNING *** CODE ADF3008:**

"Dead" element 111111 encountered in element group 22.
Note: An element is called "dead" if the same boundary condition has been applied to all nodes in this element. In order to avoid "dead" elements, split the boundary condition into two boundary conditions of the same type and apply them to different sides of elements.

This message reports a possible error found in boundary condition data. The cause of this error is essentially the same as described for message ADF2804.

***** WARNING *** CODE ADF3009:**

A boundary has been applied twice-different compressible flow conditions. The side is facing the local nodal index 33 of element 11 in element group 22. The two boundary condition labels are 44 and 55. Check if the two conditions have been applied correctly.

As a general guide in modeling of high speed compressible flows, every boundary must have one and only one boundary condition for fluid and energy equations.

This message reports a possible error found in boundary condition data in the high-speed compressible flow model. Some special boundary conditions are particularly designed for high-speed compressible flows. As a general rule, each boundary can be applied only one such boundary condition. If two conditions are applied to the

same location, a thorough understanding of these boundary conditions is required.

***** ERROR *** CODE ADF3010:**

No pressure or temperature degree of freedom in the model.
For compressible flows, both pressure and temperature degrees of freedom must be active.
Check the flow assumptions or the definitions in "MASTER" command.

This message reports an error in compressible flows. Recall that in the state equation, pressure, temperature and density are coupled. Therefore, pressure, temperature and velocity must be all active in compressible flow models.

***** ERROR *** CODE ADF3011:**

Error encountered in I/O operation.

Currently available information is:

device unit = 11;
I/O flag = 22;
length of data = 33;
I/O status = 44.

Possible causes are:

* File is too big;
* Disk space is not big enough for the current problem;
* Some working files are accidentally deleted.

This message reports an unidentified error while accessing the hard disk. Make sure that the possible causes in this message are fixed.

***** WARNING *** CODE ADF3019:**

Thermal resistance boundary condition cannot be applied to high-speed compressible flows.

The thermal resistance boundary condition is currently not available for high-speed compressible flows.

***** WARNING *** CODE ADF3020:**

Line search is not available.

This message reports an error that is related to an unavailable option. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF3021:**

Incorrect flag 11 for direct coupling of fluid-structure interactions. Data file could be wrong.

This message reports an error that is related to an unavailable option. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** WARNING *** CODE ADF3023:**

Pressure datum has been ignored since there is no pressure degree of freedom in the model. The results may be incorrect.

This message reports a potential error in the model. If you are intent selecting the option of pressure datum, you must include the pressure as a variable. Or you may remove the selection of the option.

A wrong data file may also cause this error. In this case, refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF3024:**

No pressure degree of freedom at node 11 where pressure datum is applied. Pressure datum has been ignored. The results may be incorrect.

This message reports an error in the pressure datum data. The pressure datum must be applied to a point or node that has pressure degree of freedom.

A wrong data file may also cause this error. In this case, refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF3025:**

Only the sparse solver can be used for direct coupling of fluid-structure

interactions.

This message reports an unavailable option has been selected. Only the sparse solver can be used in direct computing of two-way fluid-structure coupling problems.

***** ERROR *** CODE ADF3026:**

One-way fluid-structure interactions cannot be performed using direct coupling.

This message reports an unavailable option has been selected. Recall that the fluid solution variables are not affected by the solid displacements in one-way fluid-structure coupling problems. There is no need to use the direct computing method. Refer to the related chapters for more details.

***** ERROR *** CODE ADF3027:**

Two- and three-dimensional elements cannot be used together in this model.

This message reports an error in the selected elements. Mixed elements are generally unacceptable in the same model. Separate the 2-dimensional and 3-dimensional models.

A wrong data file may also cause this error. In this case, refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF3028:**

Adaptive mesh option cannot be applied to models that include solid element groups (e.g., conjugate heat transfer problems).

This message reports an unavailable option has been selected. Currently the mesh repair or adaptive mesh cannot be applied to conjugate heat transfer problems.

***** WARNING *** CODE ADF3029:**

Memory is not enough to keep element group data in-core. Use of adaptive mesh option requires in-core element group data.

Whenever the mesh repair or mesh adaptive option is used, we assume that the memory of the computer is enough and available during the whole computation process. Since the procedure of mesh repair is dynamic, the required memory is unable to be predicted in initiation stage of computation.

***** ERROR *** CODE ADF3030:**

Only the sparse solver can be used if adaptive mesh option is applied.

This message reports an unavailable option has been selected. Only the sparse solver can be used if mesh repair or adaptive mesh option is selected.

***** ERROR *** CODE ADF3031:**

Only triangular/tetrahedron elements can be used if adaptive mesh option is applied.

This message reports an unavailable option has been selected. The mesh repair or adaptive mesh option can only be applied to triangular 3-node or tetrahedral 4-node elements.

***** ERROR *** CODE ADF3032:**

Adaptive mesh requires in core solution. Memory is not enough to continue.

This message indicates that the memory assigned to ADINA-F is not enough. Once the mesh repair or adaptive mesh option is selected, the solution procedure is in core and, therefore, more memory may be required. Refer to the descriptions under the topic “Use of memory and disk”.

***** ERROR *** CODE ADF3033:**

Fluid-fluid or phase change boundary conditions are not applicable if adaptive mesh option is applied.

This message reports an unavailable option has been selected. The mesh repair or adaptive mesh option is currently not allowed in the problems that have fluid-fluid or/and phase-change boundary

conditions.

***** ERROR *** CODE ADF3034:**

Unsuccessful in mesh repair. Current function is "ADP_FUNCTION_NAME".

This message reports an unsuccessful attempt to repair an invalid mesh. When the solution is divergent, the mesh could be too far from being able to be repaired. Try to use small time steps, small load increments, etc. to have the solution changed smoothly.

***** ERROR *** CODE ADF3035:**

Unsuccessful in fluid model for one-way coupling (through files) of fluid-structure interaction.

This message indicates that the solution procedure in ADINA-F is unsuccessful in a separate run of one-way fluid-structure coupling problems. Test a direct run of the same problem. If it is successful, you probably had used a wrong syntax in the separate run. If it is not successful, retest your model. Refer to the suggestions under the topic "Model preparation and testing".

A wrong data file may also cause this error. In this case, refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF3036:**

Phase-change boundary conditions cannot be applied to compressible flows.

This message reports an unavailable option has been selected. In high-speed compressible flows, the phase-change boundary condition option is not available. Refer to the related chapters to see the available boundary conditions.

***** ERROR *** CODE ADF3037:**

Different element types cannot be used in a problem if it contains moving mesh boundary conditions.

This message reports an error in the selected elements. Mixed elements are generally unacceptable in the same model, particularly if

there is a moving boundary condition applied. You must also not apply special boundary conditions to lines and surfaces in the same model. Resolve these issues before rerunning this problem.

A wrong data file may also cause this error. In this case, refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF3038:**

Multi-grid solver can only be applied to problems with triangular or tetrahedron elements.

This message reports an unavailable option has been selected. The multi-grid solver can only be applied to problems with triangular 3-node or tetrahedral 4-node elements.

***** ERROR *** CODE ADF3039:**

Multi-grid solver cannot be applied to this problem since there is no pressure degree of freedom.

This message indicates that there is no pressure degree of freedom in this model. Whenever fluid velocity is considered, the pressure must be included in the model. In pure heat transfer problems where only temperature is included, use other solvers.

A wrong data file may also cause this error. In this case, refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF3040:**

Explicit method cannot be applied to problems with phase-change boundary conditions.

This message indicates a limitation of the explicit methods. These methods are generally not as stable as their implicit counterparts. You cannot use explicit methods in problems that include moving boundary

---

***** ERROR *** CODE ADF3041:**

Convection boundary conditions cannot be applied to compressible flows.

This message reports an unavailable option has been selected. In

high-speed compressible flows, the convection boundary condition option is not available. Refer to the related chapters to see the available boundary conditions.

***** ERROR *** CODE ADF3042:**

Radiation boundary conditions cannot be applied to compressible flows.

This message reports an unavailable option has been selected. In high-speed compressible flows, the radiation boundary condition option is not available. Refer to the available boundary conditions in chapter 7.

***** ERROR *** CODE ADF3043:**

Unsuccessful in solid model for fluid-structure interaction. Last status in solid model is "CALLING_STAGE".

This message indicates that the solution procedure in the ADINA solid model is unsuccessful in fluid-structure coupling problems. You can find out more information from the <solid>.log file. It is important to refer to the suggestions under the topics “Model preparation and testing” and “Formulation of fluid-structure interactions”.

If this message is printed at the first iteration in the first time step, a wrong data file may also cause this error. In this case, refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF3044:**

Unsuccessful in fluid model for fluid-structure interaction. Last status in fluid model is "CALLING_STAGE".

This message indicates that the solution procedure in the ADINA-F fluid model is unsuccessful in fluid-structure coupling problems. You can find out more information from the <fluid>.log file. It is important to refer to the suggestions under the topics “Model preparation and testing” and “Formulation of fluid-structure interactions”.

If this message is printed at the first iteration in the first time step, a wrong data file may also cause this error. In this case, refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF3045:**

Unsuccessful during iterations for fluid-structure interaction. Last status is "CALLING_STAGE".

This message indicates that the FSI iteration is divergent in an iterative computing of two-way fluid-structure coupling problems. Refer to the suggestions under the topics “Model preparation and testing” and “Formulation of fluid-structure interactions”.

If this message is printed at the first iteration in the first time step, a wrong data file may also cause this error. In this case, refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF3046:**

Unsuccessful while firstly initiate fluid model.

This message indicates that the initiation in the ADINA-F fluid model is unsuccessful. You can find out more information from the <fluid>.log file. It is likely that a wrong data file causes this error. In this case, refer to the suggestions for message ADF2111. It is also important to refer to the suggestions under the topics “Model preparation and testing” and “Formulation of fluid-structure interactions”.

***** ERROR *** CODE ADF3047:**

Unsuccessful while firstly initiate solid model.

This message indicates that the initiation in the ADINA solid model is unsuccessful. You can find out more information from the <solid>.log file. It is likely that a wrong data file causes this error. In this case, refer to the suggestions for message ADF2111. It is also important to refer to the suggestions under the topics “Model preparation and testing” and “Formulation of fluid-structure interactions”.

***** ERROR *** CODE ADF3048:**

Unsuccessful while secondly initiate fluid model.

This message indicates that the initiation in the ADINA-F fluid model is unsuccessful. You can find out more information from the <fluid>.log file. It is likely that a wrong data file causes this error. In this case, refer to the suggestions for message ADF2111. It is also important to refer to the suggestions under the topics "Model preparation and testing" and "Formulation of fluid-structure interactions".

***** ERROR *** CODE ADF3049:**

Unsuccessful while secondly initiate solid model.

This message indicates that the initiation in the ADINA solid model is unsuccessful. You can find out more information from the <solid>.log file. It is likely that a wrong data file causes this error. In this case, refer to the suggestions for message ADF2111. It is also important to refer to the suggestions under the topics "Model preparation and testing" and "Formulation of fluid-structure interactions".

***** ERROR *** CODE ADF3050:**

Current adaptive mesh cannot be applied to problems with more than one type of materials. Material type in element group 1 is 11 and material type in element group 22 is 33.

This message reports an unavailable option has been selected. Currently the mesh repair is carried out through the whole computational domain. The element groups with different materials cannot therefore be preserved.

***** ERROR *** CODE ADF3051:**

Current adaptive mesh cannot be used together with the element birth/death option. Element birth/death was found in element group 11.

This message reports an unavailable option has been selected. Currently the mesh repair is carried out for every element. The element birth/death information cannot therefore be preserved.

***** ERROR *** CODE ADF3056:**

Too many times of mesh repairs have been performed in current time step.

This message usually indicates an unsuccessful attempting on mesh repair. Too many times of repair in a time step cannot lead to a converged solution. The solution at this time step may be crucial, sensitive, going to diverge, etc. Try to use small time steps, small load increments around the current time. It is also important to refer to the suggestions under the topics "Model preparation and testing" and "Formulation of fluid-structure interactions".

***** ERROR *** CODE ADF3057:**

Unsuccessful while fluid walls are moved.

This message indicates that the solution procedure in the ADINA-F fluid model is unsuccessful in solving moving meshes. You can find out more information from the <fluid>.log file.

If this occurs before the FSI iterations, it is likely that the mesh is wrong. It could be that the big gaps existing between the interfaces of fluid and solid are too large. A wrong data file may also cause this error. In this case, refer to the suggestions for message ADF2111.

If this occurs during the FSI iterations, the structure displacements may be incorrect or too distorted. It is usually a sign that the solution procedure is divergent. It is important to refer to the suggestions under the topics "Model preparation and testing" and "Formulation of fluid-structure interactions".

***** ERROR *** CODE ADF3058:**

Unsuccessful while assemble fluid equations.

This message indicates that the solution procedure in the ADINA-F fluid model is unsuccessful in assembling the fluid equations. You can find out more information from the <fluid>.log file.

Usually a wrong data file may cause this error. Refer to the suggestions for message ADF2111. It is important to refer to the suggestions under the topics "Model preparation and testing" and "Formulation of fluid-structure interactions".

***** ERROR *** CODE ADF3059:**

Unsuccessful while assemble solid equations.

This message indicates that the solution procedure in the ADINA solid model is unsuccessful in assembling the fluid equations. You can find out more information from the <solid>.log file.

Usually a wrong data file may cause this error. Refer to the suggestions for message ADF2111. It is important to refer to the suggestions under the topics "Model preparation and testing" and "Formulation of fluid-structure interactions".

***** ERROR *** CODE ADF3060:**

Unsuccessful while fluid nodes on fluid-structure interfaces are moved.

This message indicates that the solution procedure in the ADINA-F fluid model is unsuccessful in solving moving meshes. You can find out more information from the <fluid>.log file.

If this occurs before the FSI iterations, it is likely that the mesh is wrong. It could be that the gaps existing between the interfaces of fluid and solid are too large. A wrong data file may also cause this error. In this case, refer to the suggestions for message ADF2111.

If this occurs during the FSI iterations, the structure displacements may be incorrect or too distorted. It is usually a sign that the solution procedure is divergent. It is important to refer to the suggestions under the topics "Model preparation and testing" and "Formulation of fluid-structure interactions".

***** ERROR *** CODE ADF3061:**

Unsuccessful while reset data in the solid model.

This message indicates that the solution procedure in the ADINA solid model is unsuccessful. You can find out more information from the <solid>.log file. An accidentally changing working environment can cause this problem. When there is no space left on hard disk, this error may also occur.

***** ERROR *** CODE ADF3062:**

Unsuccessful while reset data in the fluid model.

This message indicates that the solution procedure in the ADINA-F fluid model is unsuccessful. You can find out more information from the <fluid>.log file. An accidentally changing working environment can cause this problem. When there is no space left on hard disk, this error may also occur.

***** ERROR *** CODE ADF3063:**

Unsuccessful while solid solutions are being updated.

Refer to the explanations for message ADF3061.

***** ERROR *** CODE ADF3064:**

Unsuccessful while fluid solutions are being updated.

Refer to the explanations for message ADF3062.

***** ERROR *** CODE ADF3065:**

Unsuccessful while output solutions in the fluid model.

Refer to the explanations for message ADF3062.

***** ERROR *** CODE ADF3066:**

Unsuccessful while output solutions in the solid model.

Refer to the explanations for message ADF3061.

***** ERROR *** CODE ADF3067:**

Unsuccessful in the solid model during FSI iteration.

This message indicates that the solution procedure in the ADINA solid model is unsuccessful during FSI iterations. You can find out more information from the <solid>.log file. Usually this indicates that solution is divergent. It is important to refer to the suggestions under

the topics “Model preparation and testing” and “Formulation of fluid-structure interactions”.

***** ERROR *** CODE ADF3068:**

Unsuccessful in the fluid model during FSI iteration.

This message indicates that the solution procedure in the ADINA-F fluid model is unsuccessful during FSI iterations. You can find out more information from the <fluid>.log file. Usually this indicates that solution is divergent. It is important to refer to the suggestions under the topics “Model preparation and testing” and “Formulation of fluid-structure interactions”.

***** ERROR *** CODE ADF3069:**

Unsuccessful during FSI iteration.

This message indicates that the solution procedure in FSI iteration is unsuccessful. You can find out more information from the <fluid>.log and <solid>.log files. Check the stress and displacement residual history. If they indicate a converging behavior, you may increase the maximum number of FSI iterations and allow it to converge. If they indicate a diverging behavior, considering the choices below:

- Always check the solutions obtained so far to see if they are physically reasonable. Try to exclude possibilities of errors in input data.
- Reduce the stress relaxation factor if it is not too small.
- Reduce the displacement relaxation factor if it is not too small.
- It is generally suggested that tolerances in fluid and solid models are smaller than the tolerance for FSI convergence tolerance. If you have not done it, make some adjustments.
- Perform direct computing rather than the iterative computing.

It is also important to refer to the suggestions under the topics “Model preparation and testing” and “Formulation of fluid-structure interactions”.

***** ERROR *** CODE ADF4001:**

Element group type 111 (NPAR(1)) is invalid.
This could be an internal error or a wrong data file was used.

This message reports an error in the element group data. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF4002:**

Number of elements 11 (NPAR(2)) is invalid.
This could be an internal error or a wrong data file was used.

Refer to the explanations for message ADF4001.

***** ERROR *** CODE ADF4003:**

Flag 1 for the solid element group (NPAR(3)) is invalid.
This could be an internal error or a wrong data file was used.

Refer to the explanations for message ADF4001.

***** ERROR *** CODE ADF4004:**

Flag 1 for the element birth/death option (NPAR(4)) is invalid.
This could be an internal error or a wrong data file was used.

Refer to the explanations for message ADF4001.

***** ERROR *** CODE ADF4005:**

Element subtype 1 (NPAR(5)) is invalid.
This could be an internal error or a wrong data file was used.

Refer to the explanations for message ADF4001.

***** ERROR *** CODE ADF4006:**

Flag 11 (NPAR(6)) for skewed nodes is invalid.
This could be an internal error or a wrong data file was used.

Refer to the explanations for message ADF4001.

***** ERROR *** CODE ADF4007:**

Element type 1 (= NPAR(7) = number of nodes per element) is invalid.
This could be an internal error or a wrong data file was used.

Refer to the explanations for message ADF4001.

***** ERROR *** CODE ADF4008:**

Element type 1 (= NPAR(7) = number of nodes per element) is invalid.
This could be an internal error or a wrong data file was used.

Refer to the explanations for message ADF4001.

***** ERROR *** CODE ADF4009:**

Element type 1 (= NPAR(7) = number of nodes per element) is not available in two dimensions.
This could be an internal error or a wrong data file was used.

Refer to the explanations for message ADF4001.

***** ERROR *** CODE ADF4010:**

Element type 1 (= NPAR(7) = number of nodes per element) is not available in three dimensions.
This could be an internal error or a wrong data file was used.

Refer to the explanations for message ADF4001.

***** ERROR *** CODE ADF4214:**

Too many points are used in the pressure/temperature dependent material data.
number of pressure points (NCP) = 1;
number of temperature points (NCT) = 2;
maximum allowable (NCP*NCT) = 3.

Refer to the explanations for message ADF4001.

***** ERROR *** CODE ADF4215:**

Material type 1 (NPAR(15)) is invalid.
This could be an internal error or a wrong data file was used.

Refer to the explanations for message ADF4001.

***** ERROR *** CODE ADF4216:**

Number 11 (NPAR(16)) of pressure points in the pressure dependent material data is invalid.
This could be an internal error or a wrong data file was used.

Refer to the explanations for message ADF4001.

***** ERROR *** CODE ADF4218:**

Number 11 (NPAR(18)) of temperature points in the temperature dependent material data is invalid.
This could be an internal error or a wrong data file was used.

Refer to the explanations for message ADF4001.

***** ERROR *** CODE ADF4219:**

Upwinding type 1 (NPAR(19)) is invalid.
This could be an internal error or a wrong data file was used.

Refer to the explanations for message ADF4001.

***** ERROR *** CODE ADF4220:**

Bubble flag 1 (NPAR(20)) is invalid.
This could be an internal error or a wrong data file was used.

Refer to the explanations for message ADF4001.

***** ERROR *** CODE ADF4308:**

Flag 1 for clapped 9-node element (NPAR(8)) is invalid.
This could be an internal error or a wrong data file was used.

Refer to the explanations for message ADF4001.

***** ERROR *** CODE ADF4310:**

Level 1 of numerical integration (NPAR(10)) is invalid.
This could be an internal error or a wrong data file was used.

Refer to the explanations for message ADF4001.

***** ERROR *** CODE ADF4311:**

Level 1 of numerical integration in t-direction (NPAR(11)) is invalid
This could be an internal error or a wrong data file was used.

Refer to the explanations for message ADF4001.

***** ERROR *** CODE ADF4316:**

Material set number 1 (NPAR(16)) is invalid.
This could be an internal error or a wrong data file was used.

Refer to the explanations for message ADF4001.

***** ERROR *** CODE ADF4320:**

Flag 1 for energy dissipation (NPAR(20)) is invalid.
This could be an internal error or a wrong data file was used.

Refer to the explanations for message ADF4001.

***** ERROR *** CODE ADF4321:**

Too many specular nodes exist in this model.
Current 32-bit program versions can only handle problems with number of specular nodes less than 46341.

This message indicates that the size of the radiosity matrix surpasses the limit 2^{+31} on a 32-bit machine. Remember that the matrix is full and therefore its size is $8n_n^2$ bites, where n_n is the number of specular nodes. Too many specular nodes not only require large memory but also very cost in computation. Refer to the descriptions under the topic "Memory usage".

***** ERROR *** CODE ADF4322:**

Too many specular elements exist in this model.
Current 32-bit program versions can only handle problems with number of specular elements less than 46341.

This message indicates that the size of the view factor matrix surpasses the limit 2^{+31} on a 32-bit machine. Remember that the matrix is full and therefore its size is $8n_e^2$ bites, where n_e is the number of specular elements. Too many specular elements not only require large memory but also very cost in computation. Refer to the descriptions under the topic "Memory usage".

***** ERROR *** CODE ADF4323:**

PARAMETER_NAME 123456 is invalid.

This message reports an invalid integer during run time. If this occurs in the first iteration in the first time, a wrong data file may cause this error. In this case, refer to the suggestions for message ADF2111. If this occurs later in the computation, the solution may be divergent. It is also important to refer to the suggestions under the topics "Model preparation and testing" and "Formulation of fluid-structure interactions".

***** ERROR *** CODE ADF4324:**

PARAMETER_NAME 1.2345599E+18 is invalid.

This message reports an invalid real number during run time. If this occurs in the first iteration in the first time, a wrong data file may cause this error. In this case, refer to the suggestions for message ADF2111. If this occurs later in the computation, the solution may be divergent. It is also important to refer to the suggestions under the topics "Model preparation and testing" and "Formulation of fluid-structure interactions".

***** ERROR *** CODE ADF5009:**

Current iterative solver fails during iteration. Temporary variable DENA

```
or DENB 1.23450E-21 is too small. Try to use other solvers.
```

This message indicates that the iterative solver fails in the current computation.

If this occurs in the first iteration in the first time, a wrong data file may cause this error. In this case, refer to the suggestions for message ADF2111. It may also be caused by a poor matrix condition or poor initial condition. The solution may be divergent as well. For all these reasons, refer to the suggestions under the topics "Model preparation and testing" and "Strategies toward obtaining convergent solutions".

You may use the sparse solver instead.

***** ERROR/WARNING *** CODE ADF5010:**

```
The following element is overlapped:
  element (label)     =       11;
  element group number =      22;
  local nodal indices and their coordinates =
    1   .10000000E+01   .20000000E+01   .30000000E+01
    2   .40000000E+01   .50000000E+01   .60000000E+01
    3   .70000000E+01   .80000000E+01   .90000000E+01
```

This message reports an invalid or a too distorted element encountered.

If this occurs at the first iteration in the first time, make sure the following conditions are met.

- A correct data file has been used. Refer to the suggestions for message ADF2111.
- Check the reported element and its neighbor element while the element is generated. If it is overlapped, correct it.
- If the reported element is near an FSI interface, consider the possibility that the distance between the interfaces of fluid and solid models is too large. Refer to the suggestions under the topics "Model preparation and testing" and "Formulation of fluid-structure interaction".

If this occurs later in the computation, consider the following suggestions:

- Use proper leader-follower relations.

- Subdivide the domain into more regular domains that are close to being convex while the mesh is generated. Refer to the suggestions under the topic "Control of moving mesh in ALE formulation".
- Use a good initial condition in transient analyses. Refer to the suggestions under the topic "Strategies toward obtaining convergent solutions".

***** ERROR *** CODE ADF5012:**

```
The following boundary element is overlapped:
  boundary element (label)   =       11;
  boundary condition (label) =       22;
  local nodal indices and their coordinates =
   1  .10000000E+01  .20000000E+01  .30000000E+01
   2  .40000000E+01  .50000000E+01  .60000000E+01
   3  .70000000E+01  .80000000E+01  .90000000E+01
```

This message is essentially the same as message ADF5010. However, the invalid element encountered here is a boundary element. This may narrow the possible causes.

***** WARNING *** CODE ADF5013:**

```
Storage is too small to save the coefficient matrix for semi-explicit
method. Computation continues using low-speed module.
```

This message indicates the memory assigned to ADINA-F is not enough. The low-speed module of the semi-explicit method could be very slow. It is recommend using implicit methods. For example, if the sparse solver or iterative solvers are selected, the implicit time integrations are automatically enforced.

***** ERROR *** CODE ADF5014:**

```
Crashed mesh has been repaired on using the factor 1.23450E-21.
```

This message indicates the moving mesh in ALE formulation is not fully satisfied. Increments of displacement have been reduced. It is usually a sign that the solution is divergent. Refer to the suggestions under the topics "Model preparation and testing" and "Strategies toward

obtaining convergent solutions".

***** ERROR *** CODE ADF5020:**

Unsuccessful in solving wall function equation.
Possibly the computation is divergent.

This message indicates the wall functions used in two-equation turbulence models are unable to be calculated. It is usually a sign that the solution is divergent. Refer to the suggestions under the topics "Model preparation and testing" and "Strategies toward obtaining convergent solutions".

***** ERROR *** CODE ADF5021:**

Unsuccessful in bubble condensation.
Possibly the computation is divergent.

This message indicates the bubble velocity degree of freedom is unable to be condensed. If this occurs at the first iteration in the first time step, usually a zero viscosity has been defined in a fluid element group. It also could be caused by an element that is too distorted or a divergent solution. Refer to the suggestions under the topics "Model preparation and testing" and "Strategies toward obtaining convergent solutions".

***** WARNING *** CODE ADF5031:**

Improper boundary values for pressure in boundary condition 11 are specified. The A and B values are 1.23450E-21 and 2.23449E-21.

This message reports possibly invalid parameters specified in a boundary condition for high-speed compressible flows. Refer to the related chapters to see the available boundary conditions and their limitations.

***** WARNING *** CODE ADF5032:**

Improper values for temperature in boundary condition 11 are specified
The A and B values are 1.23450E-21 and 2.23449E-21.

Refer to the explanations for message ADF5031.

***** WARNING *** CODE ADF5033:**

Improper values for density in boundary condition 11 are specified.
The A and B values are 1.23450E-21 and 2.23449E-21.

Refer to the explanations for message ADF5031.

***** WARNING *** CODE ADF5034:**

Improper values for internal energy in boundary condition 11 are specified. The A and B values are 1.23450E-21 and 2.23449E-21.

Refer to the explanations for message ADF5031.

***** WARNING *** CODE ADF5035:**

Improper values for enthalpy in boundary condition 11 are specified.
The A and B values are 1.23450E-21 and 2.23449E-21.

Refer to the explanations for message ADF5031.

***** WARNING *** CODE ADF5036:**

Improper values for normal velocity in boundary condition 11 are specified. The A and B values are 1.23450E-21 and 2.23449E-21.

Refer to the explanations for message ADF5031.

***** WARNING *** CODE ADF5037:**

Improper values for flow rate in boundary condition 11 are specified.
The A and B values are 1.23450E-21 and 2.23449E-21.

Refer to the explanations for message ADF5031.

***** WARNING *** CODE ADF5038:**

Improper values for Mach number in boundary condition 11 are specified
The A and B values are 1.23450E-21 and 2.23449E-21.

Refer to the explanations for message ADF5031.

***** WARNING *** CODE ADF5041:**

Fluid flows inward at an outlet boundary. Local normal velocity and sound speed are 9.99999E-22 and 1.99999E-21 respectively. Model may be unstable or improper boundary conditions and/or initial conditions are applied.

This message reports that a flow condition becomes contrary to what the boundary condition is designed for. If this occurs at the first iteration in the first time step, probably wrong parameters have been defined in that boundary condition. If you are intent using this type of boundary condition as reported, ignore this warning. Refer to the related chapters for more details as well.

***** WARNING *** CODE ADF5042:**

Fluid flows outward at an inlet boundary. Local normal velocity and sound speed are 9.99999E-22 and 1.99999E-21 respectively. Model may be unstable or improper boundary conditions and/or initial conditions are applied.

This message reports that a flow condition becomes contrary to what the boundary condition is designed for. If this occurs at the first iteration in the first time step, probably wrong parameters have been defined in that boundary condition. If you are intent using this type of boundary condition as reported, ignore this warning. Refer to the related chapters for more details as well.

***** WARNING *** CODE ADF5043:**

Fluid flow becomes supersonic at an outlet boundary. Local normal velocity and sound speed are 9.99999E-22 and 1.99999E-21 respectively. Model may be unstable, or improper boundary conditions and/or initial conditions are applied.

This message reports that a flow condition becomes contrary to what the boundary condition is designed for. If this occurs at the first iteration in the first time step, probably wrong parameters have been defined in that boundary condition. If you are intent using this type of boundary condition as reported, ignore this warning. Refer to the

related chapters for more details as well.

***** WARNING *** CODE ADF5044:**

Fluid flow becomes supersonic at an inlet boundary. Local normal velocity and sound speed are 9.99999E-22 and 1.99999E-21 respectively. Model may be unstable, or improper boundary conditions and/or initial conditions are applied.

This message reports that a flow condition becomes contrary to what the boundary condition is designed for. If this occurs at the first iteration in the first time step, probably wrong parameters have been defined in that boundary condition. If you are intent using this type of boundary condition as reported, ignore this warning. Refer to the related chapters for more details as well.

***** ERROR *** CODE ADF6001:**

End-of-file encountered in reading the data file.
Possibly a wrong file was used.

This message reports that the data file does not contain enough information that is required. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF6002:**

No authority to use the program on this computer.

Please contact ADINA R&D, Inc.
Telephone (617)-926-5199
Telefax (617)-926-0238
Email support@adina.com

This message indicates that the license is invalid or expired.

***** ERROR *** CODE ADF6006:**

Too few records in the data file.
Possibly a wrong file was used.

This message reports that the data file does not contain enough required information. A wrong data file may cause this error. Refer to

the suggestions for message ADF2111.

***** ERROR *** CODE ADF6007:**

High-speed storage is too small to fit one column matrix data in core.

This message reports that the memory assigned to ADINA-F is far less than required. Refer to the descriptions under the topic "Memory usage" for more details.

A wrong data file may cause this error as well. In this case, refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF6008:**

No storage available to continue. Increase storage (MTOT).

Refer to the explanations for message ADF6007.

***** ERROR *** CODE ADF6009:**

Too many blocks are generated. Maximum possible number of blocks is 11

This message reports that the memory assigned to ADINA-F is far less than it required. If the Gauss elimination method (SKYLINE) has been used, change it to the sparse solver or iterative solvers. Also refer to the descriptions under the topic "Memory usage" for more details.

A wrong data file may cause this error as well. In this case, refer to the suggestions for message ADF2111.

***** ERROR/WARNING *** CODE ADF6012:**

Iteration in mass transfer is divergent. Current iteration step is 11
and the residual is 1.23450E-21.

This message indicates that the iteration in mass transfer equations is unsuccessful. The mass transfer equations are linear unless some nonlinear source terms have been applied through user-supplied material data. Check the residual history. If the tendency is convergent, increase the maximum number of iterations to allow it to converge. If the tendency is divergent, the source terms must be added in smaller increments. It is also important to refer to the suggestions under the

topics “Model preparation and testing” and “Formulation of fluid-structure interactions”.

A wrong data file may cause this error as well. In this case, refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF6014:**

Time step length 1.23450E-21 in time step block 111 is invalid.

This message indicates that a time step length is too small or negative. Check the input in the time step control.

A wrong data file may cause this error as well. In this case, refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF6015:**

No fluid-structure-interaction boundary conditions are found.
Possible errors exist in the model.

This message indicates inconsistent information. In Formulation of fluid-structure interactions, some fluid-structure interfaces must be defined in both fluid and solid models. Current fluid data file does not contain such information.

A wrong data file may cause this error as well. In this case, refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF6019:**

The sparse solver is not available on this computer.

This message indicates that the hardware sparse solver is not available from the computer vendor. You can change it to the ADINA sparse solver.

***** ERROR *** CODE ADF6022:**

Error encountered while reading data from device 1 at local index
222.

This message indicates that the program is unable to read from a device because of an unknown reason. One possible reason is that some

temporary files have been removed (not by the program). Another possibility is that a wrong data file is used. In this case, refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF6023:**

Error encountered in record "IDENTIFICATION_STRING" in restart file. Possibly a wrong restart file was used.

This message indicates that the restart file is invalid. Possible reasons are

- A wrong restart file is used.
- The file has been replaced in last execution of the problem. Remember that the restart file is replaced once the execution starts. It must always keep a backup of that file if multiple attempts may be made.
- The current version of ADINA-F is inconsistent with the one that created the restart file.
- Some limitations are not satisfied. Refer to the related chapters for more details.

***** WARNING *** CODE ADF6031:**

Norm of VARIABLE_NAME solution vector is smaller than 2.30E-14.

This message indicates that the reported solution is too small. An inadequate modeling may cause this situation. If you are intent doing so, ignore this message. It is also important to refer the suggestions under the topic “Model preparation and testing” and other related chapters.

***** WARNING *** CODE ADF6032:**

Relative residual 2.30E+12 of VARIABLE_NAME is too large.

This message indicates that the relative residual of the reported solution is too large. It could be that the solution is too small or the solution is divergent. An inadequate modeling may cause this situation. If you are intent doing so, ignore this message. It is also important to

refer the suggestions under the topic "Model preparation and testing" and other related chapters.

***** WARNING *** CODE ADF6033:**

Absolute residual 2.30E+12 of VARIABLE_NAME is too large.

This message indicates that the absolute residual of the reported solution is too large. The solution procedure is probably divergent. An inadequate modeling may cause this situation too. It is also important to refer the suggestions under the topic "Model preparation and testing" and other related chapters.

***** WARNING *** CODE ADF6037:**

No convergence is reached after the maximum number of iterations in the iterative solver. Maximum number of iterations = 11, present time step = 22 and present equilibrium iteration = 33.

This message indicates that the iterative solver is unsuccessful while the current matrix system is solved. If the rest of the iterations are successful, this message can be ignored although it is recommended to find out the reasons. The default of the maximum number of iterations (=3000) in iterative solvers may not be enough in the following situations:

- The condition of the matrix is poor (In this case, refer to the suggestions under the topic "Strategies toward obtaining convergent solutions").
- Too many equations are in the current system, usually more than half million. In this case you may increase the maximum number of iterations in iterative solvers.

The sparse solver is an alternative if desired.

***** ERROR *** CODE ADF6041:**

No convergence is reached after the maximum number of iterations in the iterative solver while solving pressure equations in semi-explicit method. Present iterations = 11 and residual = 1.23450E-21.

This message indicates that the semi-explicit method is unsuccessful for this model. Use other implicit methods, such as the sparse solver or iterative solvers.

***** ERROR *** CODE ADF6042:**

```
Incremental factor becomes too small.
  present iteration   =      11;
  present residual    = 2.00E-22;
  present factor (AL) = 1.00E-22.
```

Refer to the message ADF6041.

***** ERROR *** CODE ADF6043:**

```
Incremental factor becomes too large.
  present iteration         =      11;
  present relative residual = 2.00E-22;
  present absolute residual = 1.00E-22.
```

Refer to the message ADF6041.

***** ERROR *** CODE ADF6044:**

```
Last record in the boundary condition data block is incorrect.
Data file is wrong.
```

This message reports an error found in boundary condition data. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF6051:**

```
Inconsistency between the old and new values in "ID_STRING":
  the old value =         1;
  the new value =         2.
```

This message reports inconsistent information found from different sources, typically in restart runs.

An incorrect restart file could have been used. In this case, refer to the explanations for message ADF6023.

A wrong data file may also cause this error. In this case, refer to the

suggestions for message ADF2111.

***** ERROR *** CODE ADF6052:**

Inconsistency between the old and new values in "ID_STRING":
the old value = 2.0000000474844E-33;
the new value = 1.0000000237422E-33.

This message reports inconsistent information found from different sources, typically in restart runs.

An incorrect restart file could have been used. In this case, refer to the explanations for message ADF6023.

A wrong data file may also cause this error. In this case, refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF6053:**

Data is different in restart file and data file.
The inconsistency occurs in = "IDENTIFICATION_STRING";
the value in restart file = 11;
current value = 22.

Possible causes are:

* The input for the restart run does not match the previous run;
* Incorrect restart file has been used.

Refer to the explanations for message ADF6023.

***** ERROR *** CODE ADF6054:**

Data is different in restart file and data file.
The inconsistency occurs in = "IDENTIFICATION_STRING";
the value in restart file = 2.0000000474844E-33;
current value = 1.0000000237422E-33.

Possible causes are:

* The input for the restart run does not match the previous run;
* Incorrect restart file has been used.

Refer to the explanations for message ADF6023.

***** ERROR *** CODE ADF6055:**

```
Unable to open the file of unit 11. Currently available information is
ICREAT, IFORM, IACCES, LENGTH = 22, 33, 44, 55.
```

This message indicates that the reported file is unable to be opened. Try first to find out the file definitions corresponding to the unit number from the table under the topic "Memory usage". The file that is required may not be in the current working directory. If the file is a temporary file or unidentified from that table, probably the disk is full or you don't have the authority to write into the current directory.

A wrong data file may also cause this error. In this case, refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF6057:**

```
Inconsistency between old and new records.
  index of the record      =        11;
  the value in restart file =       22;
  current value            =       33.
Possibly a wrong restart file was used.
```

Refer to the explanations for message ADF6023.

***** ERROR *** CODE ADF6061:**

```
Gaps were found in equation numbers. Total number of equations is 11,
but the maximum equation number is 22.
This could be an internal error or a wrong data file was used.
```

This message reports an error found in boundary condition data. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF6062:**

```
Inconsistent records were found regarding number of constraint equations.
  number of constraint equations available     =     1;
  number of constraint equations in master data =    2.
This could be an internal error or a wrong data file was used.
```

This message reports an error found in boundary condition data. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF6063:**

Inconsistent records were found regarding the total number of equations.
number of fixed degrees of freedom = 1;
number of constraint equations = 2;
number of active degrees of freedom = 3;
sum of them must be equal to = 4.
This could be an internal error or a wrong data file was used.

This message reports an error found in boundary condition data. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** WARNING *** CODE ADF6064:**

Fluid model could be wrong! The factorized diagonal 1.0000E-24 of equation 11 is too small.
A constant pivot has been assigned to this equation.

This message is essentially the same as ADF2004.

***** WARNING *** CODE ADF6065:**

Fluid model could be unstable! The factorized diagonal 1.0000E-24 of equation 11 is too small.

This message is essentially the same as ADF2004.

***** WARNING *** CODE ADF6066:**

Mass-ratio model could be wrong! The factorized diagonal 1.0000E-24 at node (index) 11 is too small.
A constant pivot has been assigned to the equation at this node.

This message indicates a zero pivot encountered during the **LDU** decomposition or incomplete decomposition in mass transfer equations.

If this message appears in the first iteration in the first time step, a high possibility is that the model is ill posed. The possible reasons are as follows.

- No prescribed mass-ratio conditions or mass convection

conditions are applied if a steady-state analysis is performed. If mass convection condition is applied, the mass transfer coefficient is wrongly specified as zero. The associated time function must be checked as well.

- The mass transfer diffusion coefficient is wrongly specified as zero.
- An improper unit system is used.

If this message appears after a few iterations, possibly the iteration is divergent in the fluid model. It could be an indication that the model is improperly defined, load increments are too large, the physical phenomenon is unstable, etc. Refer to the topic "Model preparation and testing" for helpful tips.

***** WARNING *** CODE ADF6067:**

Mass-ratio model could be unstable! The factorized diagonal 1.0000E-24 at node (index) 11 is too small.

Refer to the explanations for message ADF6066.

***** WARNING *** CODE ADF6068:**

Current mesh could be invalid! The factorized diagonal 1.0000E-24 at node (index) 11 is too small.
A constant pivot has been assigned to the equation at this node.

This message indicates a zero pivot encountered during the **LDU** decomposition or incomplete decomposition in moving meshes.

If this message appears in the first iteration in the first time step, a high possibility is that mesh is invalid. In this case, check the elements around the reported node.

If this message appears after a few iterations, possibly the iteration is divergent in the fluid model. It could be an indication that the model is improperly defined, load increments are too large, the physical phenomenon is unstable, etc. Refer to the topic "Model preparation and testing" for helpful tips. Refer to the related chapters on mesh control as well.

***** WARNING *** CODE ADF6069:**

Current mesh could be invalid! The factorized diagonal 1.0000E-24 at node (index) 11 is negative.

Refer to the explanations for message ADF6068.

***** ERROR *** CODE ADF6070:**

Node (label) 22 is out of range [1,11].

This message reports an invalid nodal label encountered. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF6072:**

Skew system 111 is out of range [33,44]. This skew system has been applied to node 22.

This message reports an invalid skew system label encountered. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF6074:**

Node (label) 111 is out of range [22,33].

See message ADF6070.

***** ERROR *** CODE ADF6075:**

Nodal index 9 is out of range [1,2].

See message ADF6070.

***** ERROR *** CODE ADF6076:**

Node (label) 11 cannot be identified since it is not defined in nodal data. Data file could be wrong.

See message ADF6070.

***** ERROR *** CODE ADF6079:**

Axis has zero length in "status_WHEN_ERROR_OCCURS".

This message reports an invalid axis defined in a boundary condition. Check the reported boundary conditions and correct errors that occur. A wrong data file may cause also this error. In this case, refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF6080:**

Axis 2 has zero length in skew system 11.

This message reports an invalid skew system encountered. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF6081:**

Skew system 11 is invalid: A-axis and B-axis are not orthogonal.
dot of the two axes = 3.30E-21

This message reports an invalid skew system encountered. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF6085:**

Program is unable to find out the function (label) 11 defined in "FUNCTION_NAME".

This message reports a time function or a one-dimensional function cannot be found while it is referred. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF6086:**

Function (label) 11 in "FUNCTION_NAME" is invalid.

This message reports a time function or a one-dimensional function cannot be found while it is referred. A wrong data file may cause this

error. Refer to the suggestions for message ADF2111.

***** WARNING *** CODE ADF6087:**

```
Function (label) 11 in "FUNCTION_NAME" is invalid.
Operators related to this function have been ignored.
```

This message reports a time function or a one-dimensional function cannot be found while it is referred. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF6088:**

```
Points are out of order in time function 11.
```

This message indicates that a time function curve is invalid because some points that define the curve are out of order. Tabulated time function curves must be input in an increasing order and no repeated points. A wrong data file may cause also this error. In this case, refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF6089:**

```
Too many points exist in time functions.
  number of points in the current time function   =     1;
  maximum number of points defined in master data =     2.
```

This message reports inconsistent information existing in time functions. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF6090:**

```
time function points do not cover computational time range.
  computational time range                = [1.00E-22,2.00E+00];
  minimum and maximum time function points = [2.99E-22,4.00E-03].
```

This message indicates that the range covered by the specified time functions is smaller than the range the computation performed. Modify these time functions or modify the time step control data.

***** ERROR *** CODE ADF6092:**

Variable identification number 11 in ONE-D-FUNCTION 22 is invalid
Possible values are between 33 and 44.
This could be an internal error or a wrong data file was used.

This message reports an invalid parameter in the reported one-dimensional function. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF6093:**

Argument identification number 11 in ONE-D-FUNCTION 22 is invalid
Possible values are between 0 and 2.
This could be an internal error or a wrong data file was used.

This message reports an invalid parameter in the reported one-dimensional function. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** WARNING *** CODE ADF6094:**

An input flag indicates that the flow is dependent of mass-ratios. However, all mass expansion coefficients in the buoyant force term are zero. The input flag has been ignored.

This message reports inconsistent control in the input. Mass transfers may not affect the fluid solutions unless one of the following conditions is met:

- At least one mass expansion coefficient is nonzero.
- Fluid materials are mass-ratio dependent.

When mass transfers do not affect the fluid solutions, they do not need to be coupled with fluid solutions.

***** WARNING *** CODE ADF6095:**

An input flag indicates that the flow is independent of mass-ratios. However, non zero mass expansion coefficients in the buoyant force term are found. The input flag has been changed to indicate the mass-ratio dependency.

This message reports inconsistent control in the input. Mass transfers will affect the fluid solutions if one of the following conditions is met:

- At least one mass expansion coefficient is nonzero.
- Fluid materials are mass-ratio dependent.

In this case, the coupling of the fluid and mass-ratios must be performed.

***** ERROR *** CODE ADF6096:**

Incorrect data was found in the initial mass-ratios. Data file could be wrong. Currently available information is:
N1, N2, NN = 11, 22, 33.

This message reports an invalid record in the mass initial condition data. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** ERROR *** CODE ADF6100:**

An error message passed out from SLVMSG. The current status is: "status_WHEN_ERROR_OCCURS".

This message reports an error was encountered in the sparse solver. A wrong data file may cause this error. Refer to the suggestions for message ADF2111.

***** WARNING *** CODE ADF6106:**

Next time step length 1.00E-22 is too small.

This message reports a very small time step length has been encountered during an automatic time step cut. This step will be ignored.

***** WARNING *** CODE ADF6107:**

Explicit methods may be unstable if the time steps are larger than the critical time step length. It is suggested to use automatic time stepping with the CFL number smaller than one.

This message reports a potential unstable factor in explicit methods. When time steps are specified, they must be small enough to satisfy the CFL condition at all solution times. If the step sizes are too small, you will perform unnecessary computations. Therefore, it is recommended to use the automatic time step (CFL) option for all explicit methods. The selected CFL number must be smaller than 1, preferably around 0.8.

***** ERROR *** CODE ADF6108:**

Explicit methods cannot be used for moving-mesh problems.
(example problems are fluid-structure interaction, moving wall, etc.).

This message indicates a limitation of the explicit methods. These methods are generally not as stable as their implicit counterparts. You cannot use explicit methods in problems that include moving boundaries.

***** WARNING *** CODE ADF6109:**

Explicit methods may be unstable if the CFL number is larger than one. Current CFL number is 1.00E-22. Check your input related to automatic time stepping.

Refer to the explanations for message ADF6107.

***** WARNING *** CODE ADF6110:**

Maximum number of CFL related time step cuts has been reached without arriving at the required time.

This message indicates that the required solution at the specified time has not been obtained. This is normal if you use time integration methods with the automatic time step (CFL) option. The solution obtained in the last time step has been saved. In case a continuous transient simulation is required, you can perform a restart run.

***** ERROR *** CODE ADF6111:**

Maximum number of ATS related time step cuts has been reached without arriving at the required time.

This message indicates that the ATS method is unsuccessful. We recommend the suggestions described under the topics "Model preparation and testing" and "Strategies toward obtaining convergent solutions". Theoretically speaking, you can increase the maximum number of cuts that is allowed in the ATS method. You should know other choices as well.

***** ERROR *** CODE ADF6112:**

Solid and fluid models are not connected properly. This could occur if:
* Solid file is not specified;
* Solid file is incorrect;
* No fluid-structure interface specified in solid model (where it is called FSBOUNDARY in command);
* This is not a fluid-structure-interaction problem, but a fluid-structure interface is defined in fluid model.

Recall that fluid-structure interaction needs both fluid and solid models connected internally through fluid-structure interfaces. The interfaces are defined as FSBOUNDARY and FLUID-STUCTURE BOUNDARY-CONDITION in solid
and fluid models respectively.

This message indicates that the fluid model and solid model cannot be connected through their interfaces. Refer to the related chapters for more details on fluid-structure interactions.

***** WARNING *** CODE ADF6113:**

Convergence is obtained in one fluid-structure iteration. This may be caused by an inadequate convergence criterion. It could also occur if the model is not created properly.

This message indicates that the converged solution is obtained after one FSI iteration. If this is what you are intent modeling, ignore this message.

However, inadequate modeling frequently causes this situation. For example, the fluid stress is too small to affect the structure. If this is physically true, no coupling model is necessary. On the other hand, if the driven factor comes from the solid model, the displacement is too

small to affect fluid solutions. In this case, a one-way FSI coupling is more suitable.

***** ERROR *** CODE ADF6114:**

Automatic-time-stepping method ATS is currently not available for one-way coupling (through files) of fluid-structure iteration.

This message reports a limitation on ATS method. When one-way fluid-structure coupling analyses are performed separately, the solutions information is exchanged through the file <*solid*>.fsi. The solutions therefore must be computed as specified. ATS cut information cannot be enforced in the solid model in later runs.

***** ERROR *** CODE ADF6115:**

Explicit methods cannot be used together with ATS option.

This message indicates a limitation of the explicit methods. ATS is very unpractical if it is used in explicit methods and therefore are not allowed.

***** ERROR *** CODE ADF6116:**

Explicit methods cannot be used for mass transfer.

This message indicates a limitation of the explicit methods. These methods are generally not as stable as their implicit counterparts. It is not allowed to use explicit methods in problems that include mass transfers. You can perform the analysis using implicit methods. The sparse solver and iterative solvers will automatically enforce the implicit time integrations.

***** ERROR *** CODE ADF6117:**

Explicit methods cannot be used for steady-state analyses.

This message indicates a limitation of the explicit methods. These methods are generally not as stable as their implicit counterparts. Always perform transient analyses in explicit time integrations.

***** ERROR *** CODE ADF6118:**

Explicit methods cannot be used for turbulent flows.

This message indicates a limitation of the explicit methods. These methods are generally not as stable as their implicit counterparts. It is not allowed to use explicit methods for modeling turbulent flows.

***** WARNING *** CODE ADF6121:**

Joule-Heat model could be wrong! The factorized diagonal 1.0000E-24 at node (index) 11 is too small.
A constant pivot has been assigned to the equation at this node.

***** WARNING *** CODE ADF6122:**

Joule-Heat model could be unstable! The factorized diagonal 1.0000E-12 at node (index) 11 is too small.

These two messages report possible modeling errors in Joule-heat input. For examples, the electrical conductivity is zero or no electric potential is prescribed in the model.

***** ERROR *** CODE ADF8001- ADF8999:**

Please contact ADINA R&D, Inc.
Telephone (617)-926-5199
Telefax (617)-926-0238
Email support@adina.com

These messages report the run time status when an error is encountered. Program is unable to identify the exact cause of the error when it occurs.

Make sure that the ADINA system has been installed properly, the working environment has not been accidentally changed (file removed, directory removed, etc.), disk has enough space for the problem, correct data file has been used, correct restart file has been used in case of restart runs, both fluid and solid data files are properly generated and specified in FSI problems, etc.

A wrong data file may cause this type of error/warning message too. In this case, refer to the suggestions for message ADF2111.

Contact ADINA R&D if necessary.

Index

2D axisymmetric flow. *See* axisymmetric flow
2D planar flow. *See* planar flow
3D fluid flow, 28, 30, 31, 37, 42, 46, 48, 49, 51
absolute pressure, 41
absolute temperature, 125, 128
absorption, 126
acoustic flow, 47
active mass-ratio, 244
adaptive mesh, 357
ADINA composite scheme, 83
air bag, 46
ALE formulation, 50, 70, 71, 93, 339, 402
  differential form, 72
  integral form, 73
analysis type, 385. *See* transient *and* steady-state
angular velocity, 31, 95, 97, 100, 101, 105, 218, 223
angular velocity condition, 95, 100, 105, 218, 223
ASME steam table, 153
ATS option, 330
  maximum subdivisions, 331
Aungier Redlich Kwong model, 49
Aungier Redlich Kwong model, 160
averaged pressure, 112
averaged velocity, 44, 53
axisymmetric flow, 35, 37
Benard convection, 408
body force, 28, 40, 45
boundary condition, 33, 74, 87, 393
  angular velocity. *See* angular velocity condition
  concentrated load. *See* concentrated …
  convection. *See* convection condition
  distributed load. *See* distributed …
  external. *See* external condition
  fluid-fluid. *See* fluid-fluid interface
  free surface. *See* free surface
  FSI. *See* fluid-structure interface
  gap. *See* gap condition
  high-speed compressible, 204–29
  incompressible, 90–138
  low-speed compressible, 90–138
  mass transfer, 246–48
  mass-flow-rate, 139
  phase-change. *See* phase-change boundary
  porous medium, 170
  prescribed variable. *See* prescribed …
  radiation. *See* radiation condition
  shell-thermal, 138
  slightly compressible, 90–138
  specular radiation. *See* specular radiation
  subsonic. *See* subsonic at inlet/outlet
  supersonic. *See* supersonic at inlet/outlet
  symmetric. *See* symmetric condition
  thermal resistance, 130
  turbulence, 179, 238
  uniform flow. *See* uniform flow condition
  user-supplied, 130, 229, 248. *See* user-supplied boundary condition
  wall. *See* fixed wall and moving wall
  zero variable. *See* zero …
boundary element, 87, 88, 96, 124, 128, 213, 216, 247, 260, 262, 272, 275, 276, 277, 278, 287
Boundary friction condition, 117
boundary layer
  thickness, 76, 77
Boussinesq approximation, 40, 45

Boussinesq number, 76
Brinkman number, 78
bulk density, 64, 65, 67, 68, 69, 70, 251
bulk modulus of elasticity, 37, 46, 141, 142, 143, 172, 185
bulk velocity, 64
buoyant force, 41, 69, 76, 77
capabilities of ADINA-F, 25
Capillary number, 78
Cartesian coordinate, 28, 31
cell Peclet number, 86
centrifugal force, 34
CFL condition, 329
CFL number. *See* CFL option
CFL option, 329, 385, 413
compressibility, 37, 38, 46, 48, 55
compressible flow, 48, 75, 243, 257, 329, 382
  high-speed. *See* high-speed compressible flow
  low-speed. *See* low-speed compressible flow
computational domain, 70, 71, 84, 90, 170, 177, 179, 209, 237, 245, 257, 263, 336, 340, 341, 390, 413
concentrated force, 88, 96, 213
concentrated heat flow, 88, 123, 215
conditional loads, 161
  boundary-distance condition, 164
  diffusion load, 163
  flow resistance load, 161
  outer-iteration condition, 165
conduction, 77, 78
confined flow, 38, 46
conjugate heat transfer, 285, 335
conservative form, 28, 37
conservative variable, 29, 193, 197, 199, 202, 203, 229, 230, 241, 395
constraint condition, 334
constraint equation. *See* constraint condition
contact condition, 404
contact discontinuity, 48
continuity equation, 28, 30, 31, 34, 35, 37, 42, 45, 46, 48, 49, 51, 84, 213
control volume, 87, 193, 197, 201, 203
convection, 77, 124
  condition, 246, 247
  forced, 77, 408
  natural, 75, 76, 395
convective velocity, 72
convergence criteria in inner iteration, 314
convergence criteria in outer iteration, 308
Coriolis velocity, 32
coupled fluid-solid system. *See* fluid-structure interaction>coupled system
cylindrical coordinate, 35
Darcy's law, 44, 45
Darcy-Forchheimer equation, 44
decouple of fluid from temperature, 40
deformable structure, 70
deformation rate, 30, 36, 52, 143, 144, 145, 147
degree of freedom. *See* number of equations
density, 28, 37, 38, 40, 48, 56, 141, 143, 147, 185
  fluid, 45, 172
  solid, 45, 172
Detached Eddy Simulation (DES) model, 60
diffusion
  fluid, 78
  mass, 78
  thermal, 76, 77
  viscosity, 76
diffusivity, 126
displacement compatibility, 259
displacement criterion, 265
dissipation, 41
distributed current density load, 345
distributed heat flux, 123, 215
distributed mass flux, 244, 247
distributed normal-traction, 88, 96, 213
divergence theorem, 84
dynamic term, 82

Eckert number, 41, 77
eigenvalue, 50
eigenvector, 50
Electro-static and steady current conduction analyses, 344
element, 287, 397
  3D brick element (27-node), 294
  FCBI brick element (8-node), 298
  FCBI prism element (6-node), 298
  FCBI pyramid element (5-node), 298
  FCBI quadrilateral element (4-node), 296
  FCBI tetrahedral element (4-node), 298
  FCBI triangular element (3-node), 296
  FCBI-C element, 300
  fluid-structure interface, 271
  high-speed compressible, 201
  line, 287
  quadrilateral element (9-node), 290
  tetrahedral element (4-node), 293
  triangular element (3-node), 288
  triangular element (6-node), 291
emissivity, 125
emittance, 126
energy equation, 28, 30, 31, 34, 35, 37, 42, 45, 46, 48, 49, 51
enthalpy, 49
environment mass-ratio, 247
environment temperature, 124, 125, 128
equilibrium iteration, 268, 303, 387, 390, 400
  FSI, 266, 267, 268
  maximum number, 267, 268
Euler α-method , 83
Euler backward method, 83
Euler equations, 50
explicit time integration, 199
explosion, 46
exponent of the pressure ratio, 250
exponent of the temperature ratio, 250
external condition, 224
FCBI elements, 84, 90
FCBI-C elements, 90
Fick's law, 64, 66
fictitious time, 329
field centrifugal force, 88, 214
field centrifugal load, 97
field element. *See* element
Field friction, 154
file, 424
finite volume method, 169, 178, 245
fixed wall, 98, 217
  no-slip, 98, 217
  slip, 99, 218
fixity condition. *See* zero variable
fluid equation, 264
fluid model, 106, 260, 263, 264, 381
fluid potential, 47
fluid-fluid interface, 109
fluid-structure interaction, 36, 42, 79, 82, 257
  coupled system, 264
  direct computing, 267
  in porous media, 281, 283
  iterative computing, 266
  one-way coupling, 264, 269, 270
  thermal effect, 281, 284
  time integration, 263
  two-way coupling, 264
fluid-structure interface, 106, 229, 272
  body connection, 272
    2D C-shape, 275
    2D O-shape, 272
    3D C-shape, 277
    3D O-shape, 276
  boundary connection, 272
  corner node, 272, 275, 276, 277
  displacement interpolation, 260
  no-slip, 106, 229
  relative distance, 261
  rotational, 106, 229
  separated boundary, 272, 275, 276, 277
  slip, 106, 229
  stress interpolation, 260
  tangential, 106, 229
  unmatched meshes, 260

flux-splitting method, 196
follower. *See* leader-follower option
Fourier's law, 29
free surface, 70, 107
free surface curvature, 108
functionality
  high-speed compressible flow, 191
  incompressible flow, 80
  low-speed compressible flow, 80
  mass transfer, 243
  porous media flow, 167
  slightly compressible flow, 80
  turbulence in high-speed compressible flow, 235
  turbulence in incompressible flow, 175
  turbulence in low-speed compressible flow, 175
  turbulence in slightly compressible flow, 175
Galerkin elements, 89
Galerkin method, 84, 87
gap
  boundary condition, 110, 229
  close, 112
  discontinous solution, 110
  open, 112
  size, 112
  status, 112
Gas models, 157
Gauss elimination method, 321, 398
geometric conservation, 194
  axisymmetric, 194
  planar, 194
  three dimenisons, 194
governing equation, 28–78
  compact form, 49
  conservative form, 28
  nonconservative form, 30
Grashof number, 77
gravitational acceleration, 29, 141, 143, 172, 185
gravitational force, 29, 40, 41
heat conduct coefficient, 29, 52, 141, 142, 143, 145, 185, 230, 231, 232
  fluid, 45, 172
  laminar, 52
  solid, 45, 172
  turbulent, 52
  turbulent, 52, 53, 54, 55
heat conductivity, 50
heat convection coefficient, 124
heat flux, 28, 84, 89
heat flux condition. *See* distributed heat flux
heat transfer, 82
heat transfer equation, 85
  discrete, 86
  exact solution, 85
  numerical solution, 86
  oscillate, 86
high-speed compressible flow, 48, 49, 191, 381
hydrostatic pressure, 41, 42
ideal gas. *See* perfect gas
Ideal gas models, 158
inactive variable, 307
incompressible flow, 38, 46, 48, 79, 82, 257, 381
incremental procedure, 408
inertia force, 76
initial condition, 83, 140, 172, 185, 200, 230, 241, 248, 280, 333, 386, 387, 405, 407, 408, 411, 413, 465
inner iteration, 303, 314
internal energy, 48, 77
inviscid, 50
inviscid fluid flow, 49
isentropic condition, 38
iterative solver AMG1, 325
iterative solver AMG2, 325
iterative solvers, 325
kinetic energy, 77
kinetic pressure, 41
kinetic stress, 42
Kunz model, 351

large-eddy-simulation model, 52
latent heat, 110
leader-follower closest option, 343
leader-follower cone option, 343
leader-follower option, 339, 341
leader-follower parallel option, 342
Lewis number, 78
liquid-vapor phase change, 347
load step size. *See* time step size
low-speed compressible flow, 48, 79, 82, 381
lubrication, 371
Mach number, 38, 41, 48, 51, 76, 77
mapping file, 280, 413, 424
  format, 332
mapping solution. *See* mapping file
mass buoyant force ratio, 251
mass concentration, 64
mass conservation, 64
mass convection, 247
mass creation rate, 64, 65, 69, 251
mass density, 64, 67
mass density flux, 64
mass diffusion coefficient, 69, 249, 250, 251
mass expansion coefficient, 249, 250, 251
mass expansion coefficients, 70
mass flow rate, 64
mass flux condition. *See* distributed mass flux
mass partial density, 64
mass ratio, 65
mass transfer, 64, 79, 243
mass transfer equation, 65, 66, 67
mass-ratio, 65, 67, 69
master degree, 334
material, 398
  high-speed compressible, 230–34
  incompressible, 141–55
  low-speed compressible, 141–55
  mass transfer, 248–55
  porous medium, 172
  slightly compressible, 141–55
material curves, 155
material model
  K-ω high-Reynolds turbulence, 187
  K-ω low-Reynolds turbulence, 187
  K-ε turbulence, 185, 241
  Carreau, 143
  constant, 141, 230, 249
  DES model, 188
  large-eddy-simulation, 145
  non-Newtonian, 142–44
  porous medium, 172
  power-law, 142, 231
  pressure-temperature-dependent, 250
  RNG K-ε turbulence, 187
  second order, 144
  shear stress transport model, 188
  Southerland's formular, 231
  Spalart-Allmaras, 188
  temperature-dependent, 40, 143
  temperature-dependent power-law, 144
  time-dependent, 142
  two-layer zonal turbulence model, 189
  user-supplied, 146, 232, 250
  velocity-dependent, 249
mathematical formulation, 39
matrix condition, 329, 385, 386, 398, 407, 413, 414
memory, 419
  basic, 419
  disk, 424
  mass transfer, 422
  moving mesh, 422
  sparse solver, 421
  specular, 422
mesh solver, 306
mixing length, 147
molar concentration, 65
molar density, 66, 67
molar density flux, 66
molar-ratio, 66, 67
molecular weight, 66

momentum equation, 28, 30, 31, 34, 35, 37, 42, 44, 46, 48, 49, 51
moving boundary, 70
moving coordinate system, 97
moving mesh, 339, 341, 402
moving velocity, 72
moving wall, 70, 101, 219
  displacement, 101, 102, 104, 105, 220, 222, 223
  no-slip, 101, 220
  rotational, 105, 223
  slip, 89, 102, 220
  tangential, 104, 222
multi-grid solver. *See* solver>multi-grid
natural boundary condition, 84
Navier-Stokes equation, 28, 30, 34, 42, 51, 72, 74
Navier-Stokes equations, 73
neutral species, 64
Newton-Raphson method, 131, 169, 268, 303, 405, 408, 413
nodal interpolation, 306
noise, 46
nonconservative form, 37, 46
nondimension, 74
  coordinate reference, 74
  density scale, 74
  length scale, 74
  mass-ratio scale, 74
  reference, 74
  scale, 74
  specific heat scale, 74
  temperature reference, 74
  temperature scale, 74
  velocity scale, 74
normal-traction condition. *See* distributed normal-traction
number of equations, 424
number of specifies, 64
numerical instability, 85
numerical methods
  high-speed compressible flow, 193
  incompressible flow, 82
  low-speed compressible flow, 82
  mass transfer, 244
  porous medium, 169
  slightly compressible flow, 82
  turbulence, 177, 237
numerical oscillation, 113
one-way coupled thermal-mechanical interaction, 285
outer iteration, 303, 304, 308
Peclet number, 77, 85
Peng Robison model, 49
Peng Robison model, 160
perfect gas, 48, 77, 141, 185, 230, 241, 417
periodic boundary condition, 121
permeability, 42, 44, 172, 284
PFSI, 282, 283
phase change, 347
phase-change boundary, 109
physical stress, 42
physical traction. *See* mathematical formulation
PISO, 306
planar flow, 34
Poisson equation, 45
pore pressure. *See* pressure
porosity, 44, 172, 283
porous fluid-structure-interaction, 283
porous media, 283
porous media flow, 42, 167, 257, 381
porthole file, 424
Prandtl number, 41, 52
prescribed density, 212
prescribed electric potential, 345
prescribed mass-ratio, 246
prescribed pressure, 94
prescribed rotational velocity, 95
prescribed specific discharge, 212
prescribed tangential velocity, 104, 222
prescribed temperature, 122
prescribed total energy, 215
prescribed turbulence K-ε, 180, 181, 238
prescribed variable, 88

prescribed velocity, 93
pressure, 29, 37, 41, 45, 46, 48
pressure datum, 337
pressure interpolation, 306
pressure-dependent condition, 132
pressure-dependent material, 232
pressure-temperature-dependent material, 232
primary unit, 417
primitive variable, 29, 82, 169, 193, 202, 204, 230
P-T-dependent material, 154
radiation condition, 89, 125
radiosity, 89
radiosity equation, 89, 128
random motions of fluid, 51, 52
rarefaction, 48
ratio of specific heats, 38
ray tracing, 89
Rayleigh number, 76
RE model, 371
RE model for rough boundaries, 372
RE model for smooth boundaries, 371
real gas models, 49
Real gas models, 157
Redlich Kwong models, 159
reference datum of coordinate. *See* nondimension>coordinate reference
reference mass-ratio, 70, 249, 250
reference pressure, 250
reference temperature, 40, 141, 143, 172, 185, 231, 250
Reichardt law, 56
relative velocity, 33, 72
Relaxation factors in outer iteration, 311
restart analysis, 331, 411
Reynolds equation, 371
Reynolds number, 41, 51, 78
rigid motion of element groups, 356
RNG-based subgrid-scale model, 145
rotating frame, 31, 34
rotation center, 95, 97, 101, 105, 219, 223
rotation velocity:, 33
rotational velocity condition. *See* prescribed rotational velocity
RPBCG iterative solver. *See* solver>RPBCG
RPGMRES iterative solver. *See* solver>RPGMRES
SAM, 357
saturated porous media, 42
Schmidt number, 78
second viscosity, 29, 231
  ratio, 230
second-order scheme in Segregated method, 306
Segregated method, 304
shape factor, 125, 128
shear rate. *See* deformation rate
shear stress, 49
shear stress transport model, 56
shear viscous layer, 77
shock absorber, 46
shock wave, 48
SIMPLE, 306
SIMPLEC, 306
singular, 51
  equation, 51
  matrix, 405
skew system, 333
slave degree, 334
Sliding-mesh interface, 113
slightly compressible flow, 38, 46, 79, 82, 257, 381
Smagorinsky-Lilly subgrid-scale model, 145
Soave Redlich Kwong model, 160
Soave Redlich Kwong model, 49
solid density, 110
solid elements, 272
  2D- potential-based fluid, 272
  2D-displacement-based fluid, 272
  2D-solid, 272
  3D- potential-based fluid, 272
  3D-displacement-based fluid, 272
  3D-solid, 272

beam, 272, 274, 275
iso-beam, 272, 276
pipe, 276
pipe, 272
plate, 272
shell, 272
truss/cable, 272, 276
solid equation, 263, 264
solid model, 106, 260, 263, 264
solid velocity, 56, 72
solid wall, 182, 239
solution increment. *See* solution vector
solution vector, 130, 169, 264, 303
fluid, 264, 266, 269
solid, 264, 266, 269
solver, 398
COLSOL, 321
explicit. *See* explit time integration
in-core, 322, 324, 421
multi-grid, 325, 398
out-of-core, 322, 324, 421
RPBCG, 325, 398
RPGMRES, 325, 398
SKYLINE, 398
sparse, 322, 398
sonic surface, 48
sound speed, 38, 76, 389
Spalart-Allmaras model, 58
sparse solver. *See* solver>sparse
special boundary conditions, 48, 88, 98, 124, 216
species coupled with fluid, 245
specific heat, 141, 142, 143, 185, 230, 231, 232
at constant pressure, 48
at constant volume, 31, 48
fluid, 45, 172
solid, 45, 172
specific internal energy, 29
specific kinetic energy, 29
specific rate of heat generation, 28, 141, 142, 143, 172, 185, 230, 231, 232
specific total energy, 28, 34
specific total energy in rotating frame, 34
specular group, 128
specular radiation, 89, 126
specular reflection, 126
stability, 42, 83, 85, 132, 199, 200, 251, 264, 329, 330, 388, 408
stagnation point, 51
standard Redlich Kwong model, 49
standard Redlich Kwong model, 159
standard TFSI, 284
state equation, 29
compressible flow, 48
incompressible flow, 38
slightly compressible flow, 46
steady-state, 51, 70, 82
steady-state analysis, 385
steered adaptive mesh, 357
Stefan-Boltzmann constant, 125, 127
stiffness matrix, 321, 329
FSI, 267
stopping criteria in Segregated method, 308
storage. *See* memory
stress, 28, 34, 36, 39, 55, 84
stress criterion, 265
structural model. *See* solid model
subsonic at inlet, 227
subsonic at outlet, 228
successive substitution method, 169
supersonic at inlet, 226
supersonic at outlet, 228
surface tension, 78, 107, 108, 141, 142, 143, 172, 185
symmetric condition, 229
system unit, 74, 414, 417
dimensionless. *See* nondimension
tangential velocity, 56
temperature, 29, 48
temperature boundary condition, 122
temperature file, 424
temperature variation, 40
temperature-dependent boundary condition, 124, 125, 127

temperature-dependent material, 40, 69, 90, 143, 144, 170, 231
temporary file, 424
TFSI, 282, 284
thermal expansion coefficient, 40, 141, 142, 143, 172, 185
thermal-mechanical interaction, 284
time, 28, 31, 71, 82
time integration, 82
time step size, 388
time-dependent boundary condition, 93, 94, 98, 124, 125, 127, 180, 181, 212, 238, 246, 247, 331
time-dependent material, 82, 98
tolerance
  coincident checking, 276, 277, 278
  displacement, 265, 267, 281
  equilibrium, 387, 400
  iterative solver, 327
  steady-state, 304, 387
  stress, 265, 267, 268, 281
traction equilibrium, 259
transient analysis, 82, 385
transmisivity, 126
trapezoidal rule, 83
turbulence, 52
  dissipation rate, 52, 56
  empirical law, 56
  fluctuating velocity, 53
  kinetic energy, 52
  universal law, 56
  viscous sub-layer, 56
  wall function, 56
turbulence model
  K-ω high-Reynolds number model, 54
  K-ω low-Reynolds number model, 55
  K-ε model, 52, 55, 56
  K-ε RNG model, 53
turbulence model, 51, 52, 175, 235
  K-ω high-Reynolds number model, 54
  K-ω low-Reynolds number model, 55
  K-ε model, 52, 53, 55, 56
  K-ε RNG model, 53
  K-ε realizable turbulence model, 63
  large-eddy-simulation model, 52
turbulence model constants, 186, 187, 241
two-layer zonal turbulence model, 60
unfactorized diagonal, 407
uniform flow condition, 100
Universal Barotropic Cavitation model, 377
upwinding, 86
  finite element approach, 87
  finite volume approach, 87
  multidimensions, 87
USCS unit, 417
User-modified-materials, 155
user-supplied boundary condition. *See* boundary condition>user-supplied
user-supplied material. *See* material model>user-supplied
usual boundary condition, 88, 93, 122, 212
vapor fraction, 347
vapor-liquid phase change, 347
variation of internal energy, 31
variational form, 84
velocity, 28
velocity of moving frame, 32
velocity of species, 64
velocity strain tensor, 29
velocity-pressure coupling, 306
vent boundary condition, 117
view factor, 89
view factor matrix, 128
virtual pressure, 84
virtual temperature, 84
virtual velocity, 84
viscosity, 29, 39, 50, 52, 77, 141, 142, 143, 144, 145, 146, 172, 185, 230, 231, 232
  laminar, 52, 56
  turbulent, 52, 53, 54, 55
viscous force, 76, 77, 78

VOF method, 353
  angle condition, 354
  equations, 353
  loading, 356
  materials, 353
  surface tension, 354
Von Karman constant, 56
wave speed, 37, 38, 51
weak form of equation, 84
zero flux of turbulence, 182, 239
zero pressure, 95
zero specific discharge, 212
zero temperature, 123
zero variable, 88
zero velocity, 94
ZGB model, 351