Dynaflow™

Version 02
Release 10.A

Jean H. Prévost

Jean-H. Prévost
Department of Civil and Environmental Engineering
Princeton University
Princeton, New Jersey 08544-5263
E-mail: prevost@princeton.edu
URL: http://www.princeton.edu/~prevost

DYNAFLOW™ Version 02 Release 10.A

Date: Spring 1981
Last Update: September 2010

DESCRIPTION

DYNAFLOW™ is a finite element analysis program for the static and transient response of linear and nonlinear two- and three-dimensional systems. In particular, it offers transient analysis capabilities for both parabolic and hyperbolic initial value problems in solid, structural and fluid mechanics. There are no restrictions on the number of elements, the number of load cases, the number of load-time functions, and the number or bandwidth of the equations. Despite large system capacity, no loss of efficiency is encountered in solving small problems. In both static and transient analyses, an implicit-explicit predictor-(multi)corrector scheme is used. The nonlinear implicit solution algorithms available include: successive substitutions, Newton-Raphson, modified Newton and quasi-Newton (BFGS and Broyden updates) iterations, with selective line search options. Some features which are available in the program include:

- ✓ Multi-field/physics analysis capabilities via selective specification of multiple solution staggers.
- ✓ Multi-staggered coupled solution analysis options.
- ✓ MPI implementation options to fully exploit the architecture of parallel computers.
- ✓ Domain decomposition options to partition equations for efficient processing on parallel computers.
- ✓ Selective element reordering options applicable to unstructured as well as structured meshes in order to allow parallel and/or vector processing of elemental arrays in blocks.
- ✓ Selective specification of high- and low-speed storage allocations options.
- ✓ Direct symmetric and non-symmetric matrix column equation solvers (in-core and out of-core Crout profile solvers). Symmetric frontal solver (in-core and/or out-of-core).
- ✓ Iterative matrix equation solvers: preconditioned conjugate gradients and GMRES with diagonal and/or element-by-element Crout/LU or Gauss-Seidel preconditioning.
- ✓ Iterative matrix-free conjugate gradient and GMRES solution procedures.
- ✓ Eigenvalue/vector solution solvers including determinant search, subspace iterations and various Lanczos algorithms.
- ✓ Equation numbering optimization option to reduce bandwidth and column heights of stiffness matrix.
- ✓ Slave nodes, equivalence nodes and multi-node constraints capabilities.
- ✓ Selective specification of element-by-element implicit, explicit or implicit-explicit options.
- ✓ Selective specification of element-by-element reduced/selective integration options.
- ✓ Coupled field equation capabilities for treatment of thermosolids, saturated porous media, multi-phase flows, and piezoelectric solids.
- ✓ Arbitrary Euler-Lagrange description options for fluid and/or fluid-structure(-soil) interaction problems.
- ✓ Xfem procedures to model discontinuities, joints, shear bands and cracks growth without need for remeshing.
- ✓ Prescribed nodal and/or surface forces options.
- ✓ Prescribed nodal displacement, velocity or acceleration options.
- ✓ Prescribed arbitrary load-time functions.

DESCRIPTION (cont'd)

- ✓ Earthquake acceleration time history generation capability, for earthquake motions compatible with prescribed acceleration response spectra.
- ✓ Prescribed consistent free-field motion capability.
- ✓ Wave transmitting boundaries.
- ✓ Isoparametric data generation schemes (Cartesian, Cylindrical/Polar and Spherical).
- ✓ Element birth/death options to model addition (birth) or removal (death) of elements (material) in the physical system.
- ✓ Layout optimization analysis capabilities.
- ✓ Capability to perform constitutive experiments along prescribed stress and/or strain paths on selected material elements within the finite element mesh.
- ✓ Complete restart capabilities with options to selectively change input data.
- ✓ Fully integrated interface with the graphical pre- and post-processing program FEMGV (available from Femsys Limited, Great Britain) (http://www.femsys.co.uk/) for both workstation and PC platforms.
- ✓ Free input format mode organized into data blocks by means of corresponding macro commands.
- ✓ Fully documented user's manual (http://www.princeton.edu/~dynaflow).

The element library contains a one-dimensional, two-dimensional, and three-dimensional continuum element with axisymmetric options. An interface element, a contact element, a slide-line element with either perfect friction or frictionless conditions, a slide-line element with Coulomb friction, a truss element, a beam element, a plate/shell element, a membrane element, a boundary element and a link element are also available for two- and three-dimensional analysis.

The material library contains a linear isotropic elastic model, a linear orthotropic elastic model, a nonlinear hyperelastic model, a nonlinear Von Mises viscoelastic model, a diffusive transport model, a linear/nonlinear thermal model, a linear/nonlinear heat conduction model, a linear piezoelectric model, a Newtonian fluid model, a Von Mises elasto(-visco)-plastic model, a Drucker-Prager elasto(-visco)-plastic model, a Mohr-Coulomb (Matsuoka's) elasto(-visco)-plastic model, a Cap elasto(-visco)-plastic model, a multi-mechanism (Ishihara's) elasto-plastic model, and a family of multi-yield elasto(-visco)-plastic models developed by the author.

DYNAFLOW™ executables are available for supercomputer, workstation and PC computing platforms. Dynaflow can be obtained by signing a licensing agreement with Princeton University. A free restricted PC/Windows demo version of Dynaflow can be obtained.

REMARKS

DYNAFLOW™ is a general purpose finite element analysis program for linear and nonlinear, two- and three-dimensional, elliptic, parabolic and hyperbolic initial boundary value problems in structural, solid and fluid mechanics. Although DYNAFLOW™ can be a very powerful analysis tool, it should be emphasized that its use requires a thorough understanding of the underlying field theories used, and of the integration techniques (both in space and time) employed.

Dynaflow

User's Manual

Contents

Contents

Contents

Contents

Contents

Contents

Contents

Contents

Contents

Contents

List of Figures

List of Tables

1.0 INPUT DATA STRUCTURE

1.1 MACRO COMMANDS

Input data are organized into data blocks by means of corresponding macro commands. The macro commands are used to identify each data block. The macro command is always the first line in the data block and may be entered in uppercase or lowercase. Data associated with the data block must follow the macro command. Only data blocks pertinent to the particular analysis to be performed need be entered.

1.2 INPUT DATA OPTIONS

In many cases, the program will assign default values for input parameters not specified by the user. These default values, when applicable, are tabulated with each variable entry description.

1.3 DATA FORMAT

Input data on any given data line are entered in a free format mode. In the free format mode, the data are input as a string of numbers separated by a comma, and/or any number of blanks. Each data line is restricted not to exceed a length of 80 characters.

1.4 CONTINUATION DATA LINE

If needed, data strings may be broken onto several data lines. For that purpose, a backslash (" \ ") or a slash (" / ") need be used at the end of each data line to be continued.

1.5 COMMENT DATA LINE

Any line with the letter C in column 1 followed by at least a blank, is treated as a comment line and is ignored by the program. Any statement following the symbol '#' or '!' on any given line is also ignored.

1.6 INPUT / OUTPUT DATA FILES

DYNAFLOW main input data are contained in a main input data file. The program creates an output file called: **name.results**, where **name** is the name defined by the command DEFINE_PROBLEM name = " " in the main input data file (the default is **name** = dynaflow). In addition the program creates output files as follows:

- **TAPE90.name**: contains coordinates and nodal connectivities
- **TAPE87.name**: contains nodal dumps
- **TAPE89.name**: contains field dumps
- **TAPE88.name**: contains nodal and field time histories dumps
- **TAPE91.name**: contains material data dumps
- **TAPE93.name.nstep**: restart/backup data for step number nstep

1.7 PROGRAM EXECUTION

To enter input data and run the program using an input file, type "dynaflow.v02" followed by the name of the input data file, and by the requested program size (size in megawords), and/or the requested length of the pointer(s) array (length in kilowords). Also, an option is available to monitor printing to the logfile by selecting the time step printout frequency (step).

shell prompt > **dynaflow.v02** filename [-size = n -length = m -step=k > & logfile &]

By default n = 30, m = 50, and k=1. Alternatively, the program size and pointer length may be defined in the file dynaflow.env. The input data file: filename should be prepared according to the user's manual, and contains input commands for DYNAFLOW input and action.

1.8 UNITS

No system of units is assumed by DYNAFLOW. Rather, it is the user's responsibility to specify the input data in a *consistent* set of units of his/her choice. Overlooking this requirement is a common source of erroneous results.

In order to guide the user, input parameter dimensions are indicated when appropriate. The notation used follows the SI convention, viz., the symbols [L, M, T] are used to indicate: L = length, M = mass, T = time. Thus, for instance:

Symbol	Description	Unit	Example (SI)
x	coordinate	L	m
m	mass	M	kg
t	time	T	sec.
d	displacement	L	m
v	velocity	L / T	m / sec.
a	acceleration	L / T^2	m / $sec.^2$
ρ	mass density (per unit volume)	M / L^3	kg / m^3
b	body force (per unit mass)	L / T^2	m / $sec.^2$
g	acceleration of gravity	L / T^2	9.81 m / $sec.^2$
f	force	ML / T^2	N
E	modulus of elasticity	M / LT^2	N / m^2 (= Pa)
μ	viscosity	M / LT	Pa sec. (= Poiseuille)
k	hydraulic conductivity	L / T	m / sec.

Note/
The SI unit Poiseuille must be distinguished from the Poise (1 Poiseuille = 10 Poise). For instance, water at 20˚C has a viscosity of almost exactly 10^{-3} Pa sec.

LIST OF MACRO COMMANDS

Note	Command	Description
		PROBLEM DEFINITION
(1)	DEFINE_PROBLEM	Start New Input Record
(2)	OPEN_FILE	Open a File
	CLOSE_FILE	Close a File
	PARAMETERS	Define Parameters
	NODAL_COORDINATES	Nodal Coordinates
	NODAL_BOUNDARY_CONDITIONS	Nodal Boundary Conditions
	SLAVE_NODES	Slave Nodes
	EQUIVALENCE_NODES	Equivalence Nodes
	ALE_NODES	Euler-Lagrange Nodes
	INITIAL_D0	Initial Nodal Displacement
	INITIAL_V0	Initial Nodal Velocity
	BACKGROUND_NODAL_FIELD	Background Nodal Field
	NODAL_LOADS	Prescribed Nodal Forces / Kinematics
	SURFACE_LOADS	Prescribed Surface Forces
	CONVECTIVE_SURFACES	Convective/Radiative Surfaces
	ALE_FREE_SURFACE	Mixed Euler-Lagrange Free Surface
	READ_FREE_FIELD_MOTION	Read Motion Request
	FILTER_FREE_FIELD_MOTION	Filter Motion Request
	FREE_FIELD_NODES	Free-Field Nodal Data
	LOAD_TIME_FUNCTION	Load-Time Function
	FILTER	Load-Time Function Filtering
	DEFINE_REGION	Define a Region
(3)	DEFINE_ELEMENT_GROUP	Define a Group of Elements
	ELEMENT_GROUP	Element Group Data
	DEFINE_MATERIAL_MODEL	Define a Material Model
(4)	NODAL_CONNECTIVITY	Element Nodal Connectivity

LIST OF MACRO COMMANDS (Continued)

Note	Command	Description
	OUTPUT / PRINT REQUESTS	
	ECHO	Echo Input
	PRINT	Print Subsequent Input (Default)
	NOPRINT	Do Not Print Subsequent Input
	PRINT_ITERATIONS	Iterations Printout
	PRINT_PIVOTS	Pivots Printout
	PRINTOUT	Nodal and/or Spatial Printout
	NODAL_PRINT	Selective Nodal Printout
	NODAL_HISTORY	Nodal Time History
	NODAL_REACTION	Nodal Reaction Time History
	TRANSLATOR	Target Post-Processor
	MESH_DUMP	Mesh Dump
	NODAL_DUMP	Selective Nodal Dump
	FIELD_DUMP	Selective Field Dump
	WRITE_MOTION	Write Motion
	RECOVER_ERROR	Error Recovery
	RECOVER_STRAIN_ENERGY	Strain-Energy Recovery
	SYSTEM_COMPLIANCE	System Compliance

LIST OF MACRO COMMANDS (Continued)

Note	Command	Description
	EXECUTION REQUESTS	
	TIME_SEQUENCE	Specify Global Time Stepping
	DEFINE_STAGGER	Define a Solution Stagger
	STAGGER_CONTROL	Staggered Solution Control
(5)	INITIALIZE_D0	Initialization Request for D0
	INITIALIZE_V0	Initialization Request for V0
	INITIALIZE_A0	Initialization Request for A0
(6)	TIME_INTEGRATION	Time-Integration Parameters
	LINEAR_SOLVER	Linear Equation Solver Selection
	NONLINEAR_ITERATIONS	Iteration Requests
	EIGENVALUE_SOLUTION	Eigenvalue Solution
(7)	RECOVER_ERROR	Error Recovery
	LAYOUT_OPTIMIZATION	Layout Optimization
(8)	CONSTITUTIVE_EXPERIMENT	Constitutive Test Requests
	BACKUP	Backup Request
(9)	RUN_SOLVER	Run Solver
(10)	STOP	Stop Program
	UPDATE/CLEAR REQUESTS	
	UPDATE_COORDINATES	Update Nodal Coordinate Array
	REMESH	Remeshing Request
	CLEAR_D	Clear Nodal Displacement Array
	CLEAR_V	Clear Nodal Velocity Array
	CLEAR_A	Clear Nodal Acceleration Array
	CLEAR_STRESS	Clear Stress Array
	CLEAR_T	Reset Time to 0.0

LIST OF MACRO COMMANDS (Continued)

Notes/

(1) Define problem command serves to indicate the start of a new problem record.

(2) The input data segments can be put in any order.

(3) The elements may be read in groups (consult Chapter 9 for details).

(4) Element connectivity data must be entered as part of an element group data block (see Chapter 9 for further details).

(5) For certain problems, it is required that an initialization takes place for $\mathbf{D}_0$ (e.g., for pressure dependent materials which require that gravity induced initial stresses be first computed). (See Section 12.3.)

(6) Time-stepping and/or eigenvalue or constitutive experiment requests must be provided in order to direct the code toward an appropriate execution mode.

(7) Determine the magnitude of the error at any given time step.

(8) This option allows testing of the material constitutive modules by prescribing stress or strain paths.

(9) Run_Solver (mode = data_check or execution) must be the last command of the problem record to direct the solver toward an appropriate execution mode. In the data check mode, input data are printed out and storage requirements are indicated. *This mode should be employed prior to making expensive executions.*

(10) STOP must be used to indicate the end of the data file (alternatives are QUIT, EXIT or STOP).

LIST OF ELEMENT KEYWORDS

DYNAFLOW includes a general element library. The elements are identified by means of corresponding keywords which are used to identify each element data block.

Keyword	Description
QDC_solid	Solid Continuum (Lagrangian)
QDC_porous	Coupled Solid/Fluid Porous Continuum
QDC_pressure	Scalar Diffusion Equation
QDC_Darcy	Darcy Flow in Porous Continuum
QDC_Stokes	Stokes flow
QDC_fluid	Fluid Continuum (Eulerian)
QDC_Helmoltz	Helmoltz/Laplace equation
QDC_transport	Scalar Convection/Advection-Diffusion Transport Equation
QDC_thermal	Coupled Solid/Thermal Continuum
QDC_heat	Scalar Heat Equation
QDC_ale	Mesh Displacement Field
QDC_reservoir	Coupled Porous Solid / Two-phase Flow
QDC_flow	Multi-Phase Flow Equations
QDC_charge	Electric Charge Equation
Multi_phase_transport	Multi-Phase Transport Equation
Multi_phase_heat	Multi-Phase Heat Equation
Interface_surface	Interface w/ Coulomb Friction
Contact_surface	Nodal Contact
Slide_line	Slide Line
Slide_coulomb	Slide Line w/ Coulomb Friction
Crack_xfem	Crack/Joint
Multi_point_constraint	Multi Point Constraint
Linear_truss	Structural Truss (linear)
Nonlinear_truss	Structural Truss (nonlinear)
Linear_beam	Structural Beam/Frame (linear)
Nonlinear_beam	Structural Beam/Frame (nonlinear)
Plate	Structural Plate (linear)
Shell_plate	Structural Shell (linear)
Shell_bilinear	Structural Bilinear Shell (linear)
Membrane	Structural Membrane
Nodal_mass	Nodal Mass
Nodal_damping	Nodal Damping
Nodal_spring	Nodal Stiffness
Nodal_transmitting	Transmitting Boundary
Nodal_reaction	Boundary Element
Nodal_link	Link Element
Nodal_penalty	Nodal Penalty (selective)

LIST OF MATERIAL KEYWORDS

DYNAFLOW includes a general material library. The materials models are identified by means of corresponding keywords which are used to identify each material data block.

Keyword	Description
LINEAR_ELASTIC	Isotropic linear elastic
ORTHOTROPIC_ELASTIC	Orthotropic linear elastic
HYPERELASTIC	Hyperelastic
NEWTONIAN_FLUID	Newtonian fluid
MISES	Mises elasto-(visco)-plastic
DRUCKER_PRAGER	Drucker-Prager elasto-(visco)-plastic
MATSUOKA	Matsuoka elasto-(visco)-plastic
CAP	Cap model
MULTI_YIELD	Multi-yield elasto-plastic
ISHIHARA	Multi-mechanism elasto-plastic
MCREEP	Mises viscoelastic
PHILLIPS	Phillips model
JCR_CHALK	JCR chalk model
STRESS_1D	1D multi-yield model
HEAT_CONDUCTION	Heat conduction
SCALAR_DIFFUSION	Scalar diffusion
ELECTRIC_MODEL	Electric models

Notes . .

2.0 GLOBAL CONTROL

2.1 Define a New Problem

DEFINE_PROBLEM (new)

DEFINE_PROBLEM name = "<string>" , title = "<string>" , etc....

The command serves as the start of a new problem record. The command also serves to define the control parameters associated with the particular analysis to be performed.

Note	Variable Name	Type	Description
	Name	string	Name used to construct output results files. Name must be enclosed in quotation marks.
	Title	string	Descriptive title (optional). Title must be enclosed in quotation marks.
	Number_of_spatial_dimensions	integer	Number of spatial dimensions ≥ 1 and ≤ 3

EXAMPLE

```
Define_problem  /
  name = "problem_name"   /
  title = "problem_title"
```

2.2 Restart a Problem

DEFINE_PROBLEM (restart)

DEFINE_PROBLEM name = "<string>" , title = "<string>" , etc....

The command serves as the start of a new problem record. In the restart mode, existing data may be modified and/or new data may be entered.

Note	Variable Name	Type	Default	Description
	Name	string	[none]	Name used to construct output results files. Name must be enclosed in quotation marks.
	Title	string	[last]	Descriptive title (optional). Title must be enclosed in quotation marks.
	Restart_name	string	[none]	Name of problem_name defined by previous run. Name must be enclosed in quotation marks.
	Restart_at_time_step	integer	[0]	Step number at which restart is to be initiated.

EXAMPLE

```
Define_problem  /
   name = "new_problem_name"   /
   title = "new_problem_title"  /
   restart_name= "old_problem_name"  /     # Problem name to be restarted
   restart_at_time_step = ns               # Step number at which the restart is
                                             to be initiated.
```

2.3 Backup a Problem

BACKUP

BACKUP Step_number = ns , etc....

The command is used to backup current results in a restart file.

Note	Variable Name	Type	Default	Description
	Step_number	integer	[0]	Step number at which backups are to be initiated
	Backup_freq	integer	[0]	Backup frequency ≥ 0
	Backup_save	integer	[0]	Number of backup files to be saved ≥ 1 = 0, save all backup files
	Final_time on / **off**	list	[off]	Request a backup at end of run

EXAMPLE

```
Backup  /
  Step_number = ns  /          # Backup starting at step ns
  Backup_frequency = freq /    # Backup every freq time step
  Backup_save = nsave          # Save nsave backup files
```

2.4 Time Sequencing

TIME_SEQUENCE

TIME_SEQUENCE number_of_time_steps = nts , etc...

Specify global controls for time stepping and convergence.

Note	Variable Name	Type	Default	Description
• *Analysis Duration*				
	Max_number_of_time_steps	integer	[NTS]	Maximum number of time steps; NTS_max
(1)	Number_of_time_steps	integer	[0]	Number of time steps NTS: > 0 and ≤ NTS_max
	Time_at_start	real	[0.0]	Time at start of analysis ≥ 0
(1)	Final_time	real	[*]	Time at which analysis is to end unless it ends earlier due to other criteria (e.g., maximum number of time steps, or steady state convergence).
(2)	Time_step	real	[1.0]	Initial global time step increment size Dt1 > 0.0
	Time_unit **seconds** minutes hours days months years	list	[*]	Time unit
• *Time Step Control*				
(3)	Variable_time_step on / **off**	string	[off]	Variable time step option
(4)	Convergence_index	real	[1.0]	Convergence index
	Load_time_function	integer	[0]	Time step load-time function number ≥ 0
	Multiplier_time_step	real	[1.0]	Time step multiplier
	Iteration_threshold	integer	[0]	Iteration threshold below which time step multiplier is activated

(cont'd)

(cont'd)

Note	Variable Name	Type	Default	Description
	Max_time_step	real	[Dt1]	Maximum time step increment size
	Min_time_step	real	[0.0]	Minimum time step increment size ≥ 0.0
• *Recomputing Time Step*				
(4)	Redo_on_divergence on / **off**	string	[off]	Redo time step on divergence
(5)	Divergence_threshold	real	[0.0]	Divergence threshold (≤ 1.0 and ≥ -1)
	Max_number_of_redo	integer	[1]	Maximum number of successive time step increment reductions if time step is repeated after failing to converge (≥ 1).
	Redo_decrease_factor	real	[0.25]	Factor used to reduce time step increment if time step is repeated after failing to converge. (> 0 and ≤ 1.0)
• *Steady-State Options*				
	Exit_on_steady_state on / **off**	list	[off]	Select whether or not to end the run based on steady state exit criteria.
	Steady_state_freq	integer	[1]	Frequency of steady-state checks ≥ 0 (e.g., every n time step)

EXAMPLE

```
Time_sequence  /
   Number_of_time_steps = 10 /      # number of time steps to be performed
   Time_step = 1.E-2                # time step Dt = 0.01
```

Notes/

(1) The number of time steps and/or the final time are exit criteria to end the analysis. An analysis can end sooner due to other criteria (e.g., an exit on steady-state criterion). However, an analysis ends after the maximum number of time steps, or the final time if specified, whether or not a steady-state solution has been reached.

(2) The time step must be sufficiently small to accurately characterize the non-linear behavior, and to meet stability requirements (if applicable).

(3) Variable time steps can be obtained by using a corresponding load-time function, or a time step multiplier.

(4) A convergence indicator is used to evaluate how well the time steps are converging. This indicator is defined as:

$$\text{converge\_index} = \frac{\log_{10}(\text{conv\_r})}{\log_{10}(\text{tol\_r})} \le 1$$

where tol_r = convergence tolerance for the residual,

$$\text{conv\_r} = \left|\boldsymbol{r}^{(\text{i})}\right| \Big/ \left|\boldsymbol{r}^{(0)}\right| \qquad \boldsymbol{r}^{(\text{i})} = \text{residual at iteration (i)}$$

and $||$ = Euclidean norm (see Section 12.3 for more details). If the convergence indicator is less than zero the time step is diverging, whereas if it is greater than zero and less than one, the time step is not fully converged.

(5) By selecting the divergence_threshold the user may control when time steps not fully converged are to be redone by decreasing the time step increment.

2.5 Echo

ECHO

ECHO <string>

The command is used to display a message on the screen, or to send it to the log file, during a solver input phase.

Note	Variable Name	Type	Description
	No name	string	Message to be displayed can be any number of characters (≤ 80)

2.6 Print

PRINT

PRINT

The command is used to request a comprehensive output in the result file of the input data. The command may be inserted at any place within the input file.

2.7 Noprint

NOPRINT

NOPRINT

The command is used to suspend output of the input data (the default is print). The command may be inserted at any place within the input file.

2.8 Open File

OPEN_FILE

OPEN_FILE	file_name = "<string>"....

Open the file named "<string>". The file name must be enclosed in quotation marks. The command may be used during the input phase to direct the solver to read an entire sequence of input commands in a file distinct from the main input data file. At the end, the file must be closed using the CLOSE_FILE command, to redirect the solver toward the main input data file.

2.9 Close File

CLOSE_FILE

CLOSE_FILE	file_name = "<string>"....

Close the file named "<string>". The file_name is optional.

2.10 Parameters

PARAMETERS

PARAMETERS a1 = etc...

The use of parameters can be used to simplify the input data required to define a problem. Data may be specified as a single character parameter (e.g., a, b through z), two character parameters (e.g., aa, ab, …), a character and a numeral (e.g., a1 through a9), etc., up to three symbols. The only restrictions are that the data must start with a character parameter and only contain character and numeral symbols. All alphabetic input characters are automatically converted to lower case, hence there are 35,594 unique parameters permitted at any one time. Parameters may have their values redefined as many times as needed by using the PARAMETERS command.

EXAMPLE

```
Parameters /
   a1 = 1.0, a2 = 2.0 /
   b12 = 5, …
```

2.11 Run Solver

RUN_SOLVER

	RUN_SOLVER mode = etc...
	Direct the solver DYNAFLOW towards an appropriate execution mode.

Note	Variable Name	Type	Default	Description
(1)	Mode	list	[data_ check]	Type of execution mode
	data_check			Data check
	execution			Perform analysis

Notes/
(1) The Run_Solver command must be the last command of the problem record and is used to direct the solver towards an appropriate execution mode. In the data_check mode, input data are read in and printed out if requested, and storage requirements are indicated. This mode should be employed before making expensive executions.

2.12 Stop

STOP

	STOP
	Exit the input file, and ignore any subsequent commands. Alternatives are QUIT, EXIT, or STOP. The program also exits when it finds the end of file mark in the input file.

3.0 OUTPUT OPTIONS

3.1 Target Post-Processor Selection

TRANSLATOR

TRANSLATOR output_format = "<string>"

The command serves to generate output files directly usable with the selected target post-processor.

Note	Variable Name	Type	Default	Description
	Output_format IRIS_inventor SGI_solidview Femsys_femgv Spectrum_visualizer Pro_e	string	[none]	Name of target post-processor. Name must be enclosed in quotation marks.

EXAMPLE

```
Translator /
  output_format = "Femsys_femgv"
```

3.2 Nodal and/or Spatial Printout Requests

PRINTOUT

	PRINTOUT	Nodal_step = ... , etc.

Note	Variable Name	Type	Default	Description
	Nodal_step	integer	[0]	Step number at which nodal printouts are to be initiated
(1)	Nodal_freq	integer	[0]	Nodal printout frequency
	Spatial_step	integer	[0]	Step number at which spatial printouts are to be initiated
(2)	Spatial_freq	integer	[0]	Spatial printout frequency
	Reaction_step	integer	[0]	Step number at which nodal reaction printouts are to be initiated
(3)	Reaction_freq	integer	[0]	Nodal reaction printout frequency
	Summary_step	integer	[0]	Step number at which nodal summary printouts are to be initiated
(4)	Summary_freq	integer	[0]	Summary printout frequency
	Final_time on / **off**	list	[off]	Select to perform a nodal and/or spatial printout at end of run (i.e., the final time)

Notes/
(1) Printed nodal output (i.e., displacements, velocities and accelerations) will occur after each multiple of nodal_freq step (see Section 3.3 for selective prints).

(2) Printed spatial output (e.g., element stresses) will occur after each multiple of spatial_freq step.

(3) Printed nodal reactions will occur after each multiple of reaction_freq step.

(4) Printed nodal summaries will occur after each multiple of summary_freq step.

3.3 Selective Nodal Printout Requests

NODAL_PRINT

	NODAL_PRINT	node1, node2, ng	

Note	Variable Name	Type	Default	Description
(1)	NODE1	integer	[0]	Node number 1 for nodal printouts
	NODE2	integer	[0]	Node number 2 for nodal printouts
	NG	integer	[1]	Generation increment

Notes/
(1) Nodal printouts (i.e., displacements, velocity and acceleration) for node numbers "node1" to "node2" at increments of "ng" will occur (see Section 3.2). If "node2" is not specified, only the value of "node1" is output. If that command is not used the nodal output of all nodes will be provided.

3.4 Mesh Dump Requests

MESH_DUMP

	MESH_DUMP	Nodal_step = ... , etc.

Note	Variable Name	Type	Default	Description
(1)	Nodal_step	integer	[0]	Step number at which nodal dumps are to be initiated
	Nodal_freq	integer	[0]	Nodal dump frequency
	Spatial_step	integer	[0]	Step number at which spatial dumps are to be initiated
	Spatial_freq	integer	[0]	Spatial dump frequency
	Nodal_time	real	[0.0]	Time at which nodal dumps are to be initiated.
	Nodal_inc	real	[0.0]	Nodal dump time increment
	Spatial_time	real	[0.0]	Time at which spatial dumps are to be initiated
	Spatial_inc	real	[0.0]	Spatial dump time increment
	Final_time on / **off**	list	[off]	Select to perform a mesh dump at end of run (i.e., the final time)

Notes/
(1) For elliptic boundary value problems the velocity vectors are actually the displacement increments computed over the last time step.

3.5 Selective Nodal Dump Requests

NODAL_DUMP

	NODAL_DUMP	nstep1, nstep2, ng	

Note	Variable Name	Type	Default	Description
(1)	NSTEP1	integer	[0]	Step number 1 for nodal dumps
	NSTEP2	integer	[0]	Step number 2 for nodal dumps
	NG	integer	[1]	Generation increment

Notes/
(1) Nodal dumps (i.e., displacements, velocity and acceleration) for step numbers "nstep1" to "nstep2" at increments of "ng" will occur. If "nstep2" is not specified, only the values at "nstep1" are output.

3.6 Selective Field Dump Requests

FIELD_DUMP

	FIELD_DUMP nstep1, nstep2, ng

Note	Variable Name	Type	Default	Description
(1)	NSTEP1	integer	[0]	Step number 1 for field dumps
	NSTEP2	integer	[0]	Step number 2 for field dumps
	NG	integer	[1]	Generation increment

Notes/

(1) Field dumps (e.g., stresses) for step numbers "nstep1" to "nstep2" at increments of "ng" will occur. If "nstep2" is not specified, only the values at "nstep1" are output.

3.7 Selective Time Nodal Dump Requests

TIME_NODAL_DUMP

TIME_NODAL_DUMP
time_1, time_2, …
< etc..., terminate with a blank record >

Note	Variable Name	Type	Default	Description
	time_i	real	[0.0]	Times at which nodal dump is requested

3.8 Selective Time Spatial Dump Requests

TIME_FIELD_DUMP

TIME_FIELD_DUMP
time_1, time_2, …
< etc..., terminate with a blank record >

Note	Variable Name	Type	Default	Description
	time_i	real	[0.0]	Times at which spatial dump is requested

3.9 Nodal Time History Requests

NODAL_HISTORY

NODAL_HISTORY displacement = on, etc…
node, ng, (ntemp(i), i = 1, ndof)
< etc..., terminate with a blank record >

Plots of nodal time histories (e.g., displacement, velocity, etc.) may be obtained.

Note	Variable Name	Type	Default	Description
	Displacement **on** / off	list	[on]	Displacement
	Velocity **on** / off	list	[on]	Velocity
	Acceleration **on** / off	list	[on]	Acceleration
	Reaction on / **off**	list	[off]	Reaction
	Coordinate on / **off**	list	[off]	Coordinate

Nodal history output data must follow.

Note	Variable Name	Type	Default	Description
(1)	NODE	integer	[0]	Node number ≥ 1 and ≤ NUMNP
(2)	NG	integer	[0]	Generation increment ≥ 0
	NTEMP(1)	integer	[0]	Degree of freedom number ≤ NDOF
	NTEMP(2)	integer	[0]	Degree of freedom number ≤ NDOF
	.	.		.
	etc.	.		.
	.	.		.
	NTEMP(NDOF)	integer	[0]	Degree of freedom number ≤ NDOF

Notes/
(1) Nodal history output data must be input for each node at which the time history of one or more degrees of freedom is to be plotted. *Terminate with a blank record.*

(2) Nodal history output data can be generated by employing a two record sequence as follows:

Record 1: LODE,LG,LTEMP(1),...,LTEMP(NDOF)
Record 2: NODE,NG,NTEMP(1),...,NTEMP(NDOF)

The output history codes of all nodes

LODE+LG, LODE+2*LG,..., NODE - MOD(NODE-LODE, LG)

(i.e., less than NODE) are set equal to those of node LODE. If LG is blank or zero, no generation takes place between LODE and NODE.

3.10 Iterations Printout Requests

PRINT_ITERATIONS

	PRINT_ITERATIONS	file_name = "<string>"

Note	Variable Name	Type	Default	Description
(1)	File_name	string	[none]	Name of file to contain iterations printouts Name must be enclosed in quotation marks

Notes/
(1) This option allows the iteration printouts to be made on a file separate from the main output file. The default (i.e., filename is left empty) assumes that the printout is contained in the main output file.

3.11 Pivots Printout Requests

PRINT_PIVOTS

	PRINT_PIVOTS	step_number = ... , etc.

Note	Variable Name	Type	Default	Description
	Step_number	integer	[0]	Step number at which pivots printouts are to be initiated.
	Frequency	integer	[0]	Number of time steps between pivots printouts.

4.0 NODAL COORDINATE DATA

NODAL_COORDINATES

NODAL_COORDINATES generation_type = type, etc...
n, numgp, (x(i, n) , i = 1 , nsd)
< etc..., terminate with a blank record >

Create the list of nodal coordinates. Two options are available. The coordinates may be read in directly as a list (optionally from another file), or may be generated.

Note	Variable Name	Type	Default	Description
(1)	Generation_type **list** Cartesian Cylindrical Spherical Lattice	list	[*]	Generation case
(2)	Reference_coordinates x_0 , y_0, z_0	real	[origin]	Reference coordinates (if needed)
	Reference_direction_axes n_x(1), n_y(1), n_z(1) n_x(2), n_y(2), n_z(2) n_x(3), n_y(3), n_z(3)	real	[ref. axes]	Reference direction axes (if needed)
(3)	Scaling_factor	real	[1.0]	Scaling factor
	File_name	string	[none]	File name (optional). Name must be enclosed in quotation marks.

- *No Generation (List) Case*
  Nodal coordinates must follow in the form:
  < Node_number , X_coordinate , Y_coordinate , Z_coordinate >
  < etc..., terminate with a blank record >
- *Generation Case*
  Nodal Coordinate Generation Data must follow.
  < terminate with a blank record >

EXAMPLE

```
Nodal_coordinates /
  file_name = "coordinate_file"   # read coordinates from file: coordinate_file
```

EXAMPLE

```
Nodal_coordinates /
  Generation_type = cylindrical /          # Select cylindrical generation
  Reference_coordinates /
    X_0 = 0.0  Y_0 = 0.0 /                 # Reference point at origin
  Reference_direction_axes /
    n_x(1) = 1.0 , n_y(1) = 0.0 /          # Reference directions coincide
    n_x(2) = 0.0 , n_y(2) = 1.0            # with reference axes in this case
  Nodal Coordinate Generation Data follow
```

Notes/

(1) In the case of Generation_type = list, no generation is used and the nodal coordinates are entered in the same format as in the file method.

(2) The default is $\mathbf{x}_0 = \mathbf{0}$, and $\mathbf{n}_1 = \mathbf{e}_1 = \{1, 0, 0\}$, $\mathbf{n}_2 = \mathbf{e}_2 = \{0, 1, 0\}$, and $\mathbf{n}_3 = \mathbf{e}_3 = \{0, 0, 1\}$ where $[\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3]$ is the triad of unit base vectors used for the global reference rectangular Cartesian axes. The coordinates are first computed as $\underline{\mathbf{x}}$ in the set of coordinate axes defined by the reference direction axes $\{\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3\}$, and thereafter referred to the global coordinate axes as $\mathbf{x}$ via the following translation and rotation:

$$\mathbf{x} = \mathbf{x}_0 + \mathbf{R} \cdot \underline{\mathbf{x}}$$

where $\mathbf{R} = [\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3]$, and $\mathbf{x}_0$ = reference point. Note that the reference direction vectors $\mathbf{n}_I$ (I =1, nsd) are restricted to be orthogonal to each other, viz.,

$$\mathbf{n}_i \cdot \mathbf{n}_j = \delta_{ij}$$

(3) The coordinate may be scaled as:

$$\mathbf{x}_j(i) = \text{scaling\_factor} * \mathbf{x}_j(i) \qquad (i = 1, \text{Numnp}; \quad j = 1, \text{Nsd})$$

4.1 Cartesian Coordinates Generation

4.1.1 Cartesian Nodal Coordinate Data

Note	Variable	Default	Description
(1)	N	[0]	Node number ≥ 1 and ≤ NUMNP
(2)	NUMGP	[0]	Number of generation points ≥ 0 = 0, no generation > 0, generate data
	X(1, N)	[0.0]	X1-coordinate of node N
	X(2, N)	[0.0]	X2-coordinate of node N
	X(3, N)	[0.0]	X3-coordinate of node N

Notes/
(1) The coordinates of each node must be defined, but need not be read in order. If the coordinates of node N are input and/or generated more than one time, the last values take priority. *Terminate with a blank record.*

(2) If NUMGP is greater than zero, this record initiates an isoparametric data generation sequence. Records 2 to NUMGP of the sequence define the coordinates of the additional generation points (see Section 4.1.2). The final record of the sequence defines the nodal increment information (see Section 4.1.3). After the generation sequence is completed, additional nodal coordinate records, or generation sequences, may follow.

The generation may be performed along a line, over a surface, or over a volume. A description of each of these options is given hereafter.

A. Generation Along a Line

The line may be defined by 2, or 3, generation points (see Figure 4.1.1), and the physical space may be 1, 2, or 3 dimensional.

In the case NUMGP = 2, linear interpolation takes place resulting in equally spaced nodal points.

In the case NUMGP = 3, quadratic interpolation is employed and graded nodal spacing may be achieved by placing the third generation point (J = 3) off center. Note that the third generation point does not generally coincide with any nodal point. The spacing in this case may be determined from the following mapping:

$$\mathbf{x}_A = \mathbf{x}(\xi_A) = \frac{1}{2}\zeta_A(\zeta_A - 1)\mathbf{x}_1{}^g + \frac{1}{2}\zeta_A(\zeta_A + 1)\mathbf{x}_2{}^g + (1-\zeta_A)(1+\zeta_A)\mathbf{x}_3{}^g$$

where ζ_A is the location of node number A in ζ-space (the nodes are placed at equal intervals in ζ-space); $\mathbf{x}_1{}^g$, $\mathbf{x}_2{}^g$ and $\mathbf{x}_3{}^g$ are the coordinates of the three generation points in **x**-space; and $\mathbf{x}_A$ denotes the coordinates of the A^{th} node in **x**-space (see Figure 4.1.2).

B. Generation Over a Surface

The surface may be defined by 4, or 8, generation points (see Figure 4.1.3) and the physical space may be 2, or 3 dimensional. In the 3-dimensional case, the surfaces may be curved.

In the case NUMGP = 4, bilinear interpolation is employed, resulting in equally spaced nodal points along generating lines.

In the case NUMGP = 8, biquadratic "serendipity" interpolation is employed and graded nodal spacing may be achieved by placing generation point 5-8 off center. Note that generation points 5-8 do not generally coincide with any nodal points. The spacing of the nodal points may be determined from the serendipity mapping.

C. Generation Over a Volume

The volume is brick shaped and may be defined by 8, or 20, generation points (see Figure 4. 1.4). In this case the physical space must be 3-dimensional.

If NUMGP = 8, trilinear interpolation is employed, resulting in equally spaced nodal points along generating lines.

If NUMGP = 20, triquadratic serendipity interpolation is employed and graded nodal spacing may be achieved by placing generation points 9-20 off center. Note that generation points 9-20 do not generally coincide with any nodal points. The spacing of the nodal points may be determined by the serendipity mapping.

4.1.2 Generation Point Coordinate Data (NUMGP-1)

The coordinates of each generation point are defined by a generation point coordinate record. The records must be read in order (J = 2, 3,..., NUMGP) following the nodal coordinate record which initiated the generation sequence (J = 1). A nodal record (see Section 4.1.3), which completes the sequence, must follow the last generation point record.

Note	Variable	Default	Description
	M	[0]	Node number
	MGEN	[0]	Generation parameter = 0, coordinates of the J^{th} generation point are input on this record; M is ignored. = 1, coordinates of the J^{th} generation point are set equal to coordinates of node M which was previously defined; coordinates on this record are ignored
	TEMP(1, J)	[0.0]	X1-coordinate of generation point J
	TEMP(2, J)	[0.0]	X2-coordinate of generation point J
	TEMP(3, J)	[0.0]	X3-coordinate of generation point J

4.1.3 Nodal Increments Data

Note	Variable	Default	Description
	NINC(1)	[0]	Number of nodal increments for direction 1
	INC(1)	[0]	Node number increment for direction 1
(1)	NINC(2)	[0]	Number of nodal increments for direction 2
	INC(2)	[0]	Node number increment for direction 2
(1)	NINC(3)	[0]	Number of nodal increments for direction 3
	INC(3)	[0]	Node number increment for direction 3

Notes/
(1) Each option is assigned an option code (IOPT) as follows:

IOPT	*Option*
1	Generation along a line
2	Generation over a surface
3	Generation over a volume

IOPT is determined by the following logic:

```
IOPT = 3
IF(NINC(3) = 0)   IOPT = 2
IF(NINC(2) = 0)   IOPT = 1
```

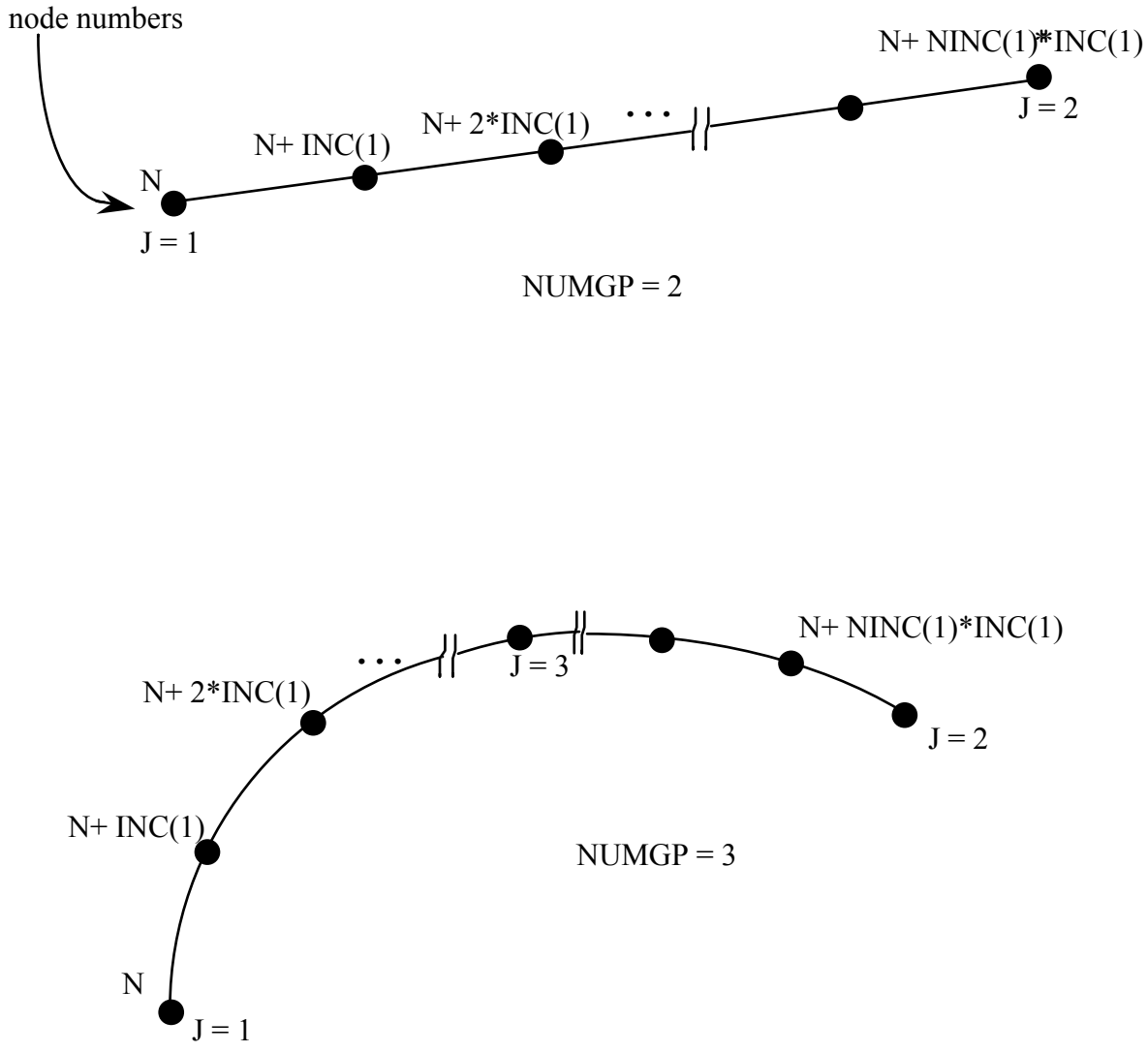


Figure 4.1.1 Nodal Generation Along a Line

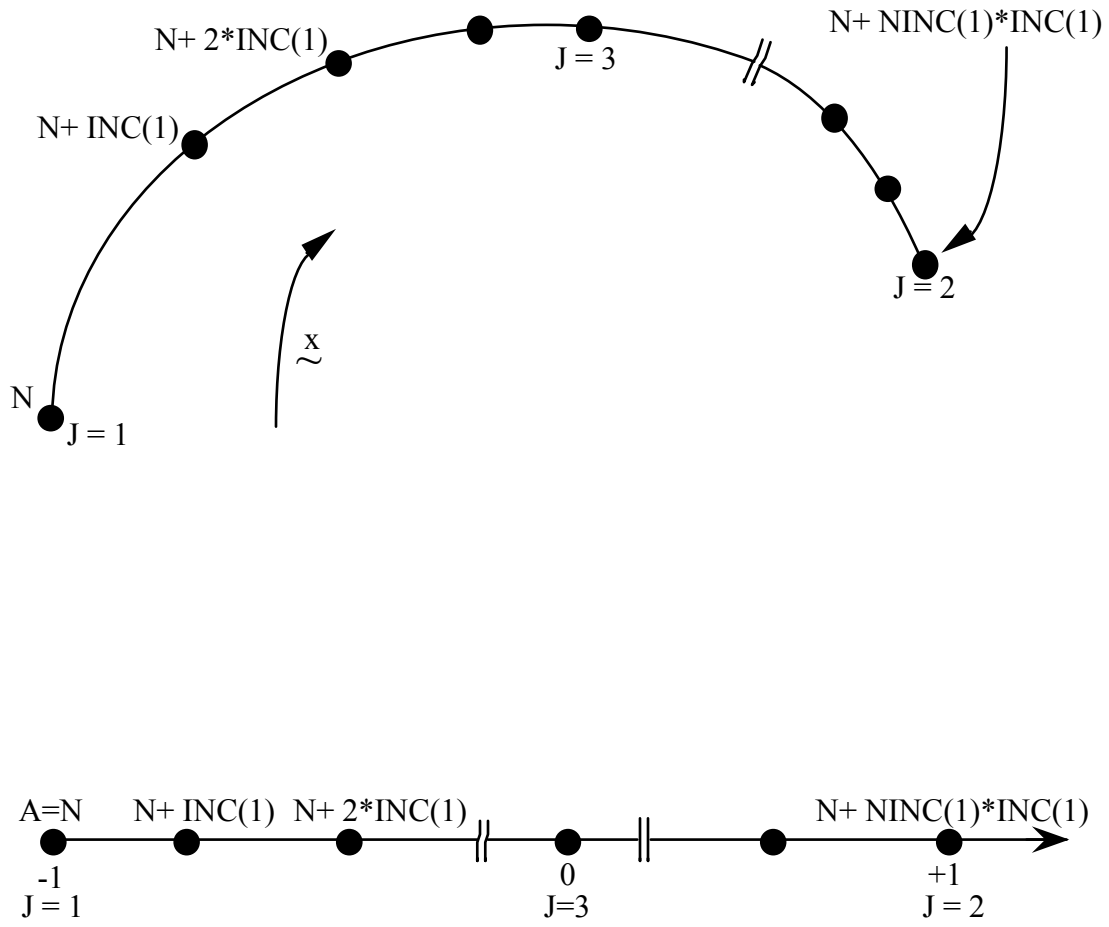


Figure 4.1.2 Nodal Generation Along a Line: Mapping from Local Interval to Physical Space

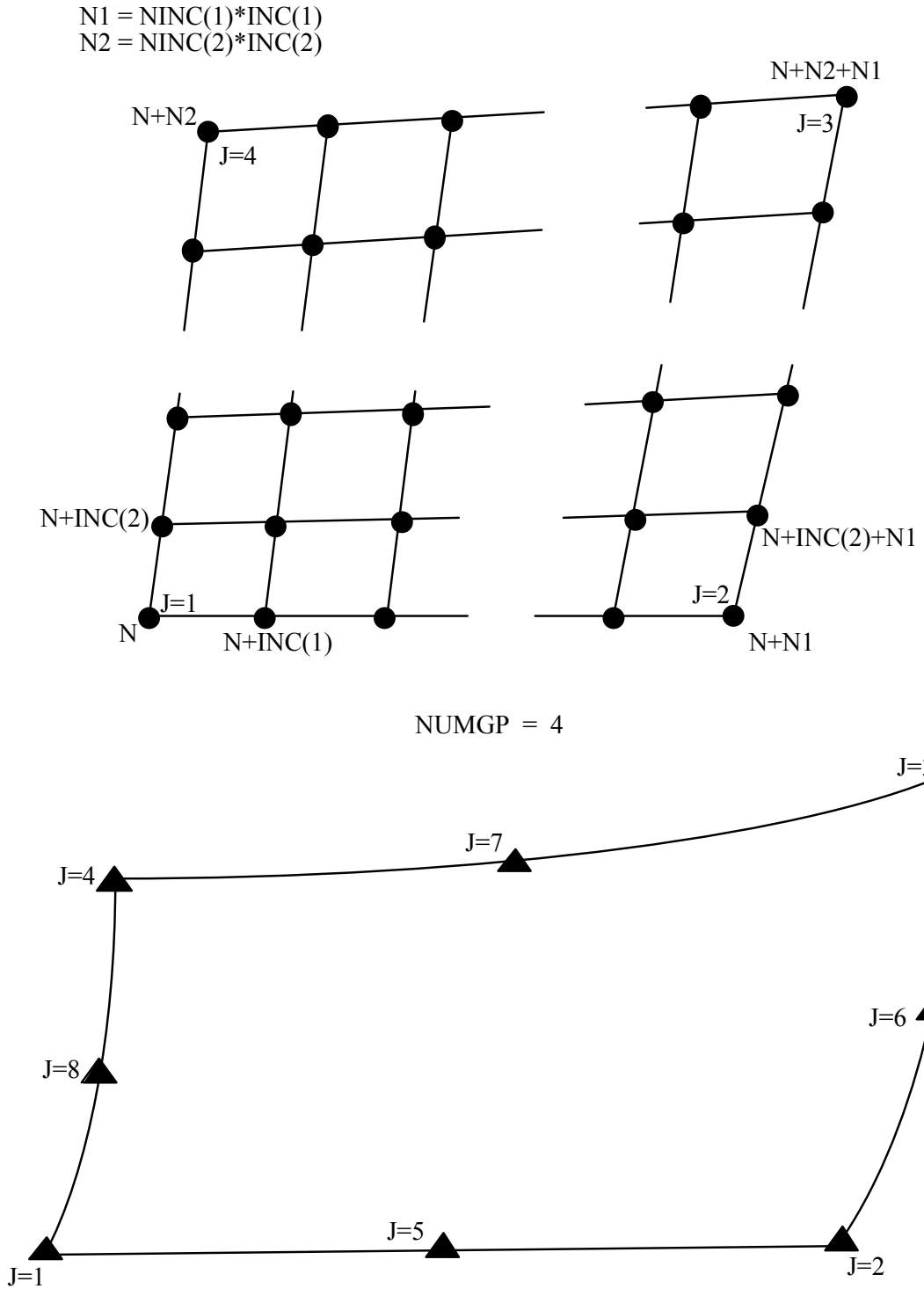
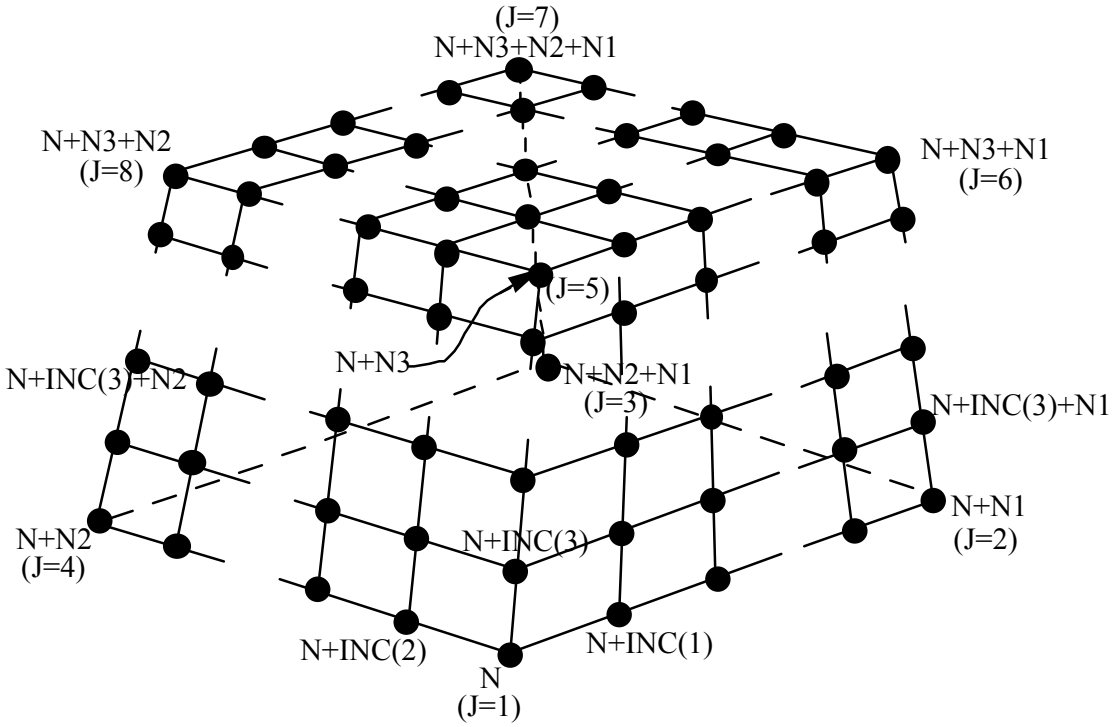
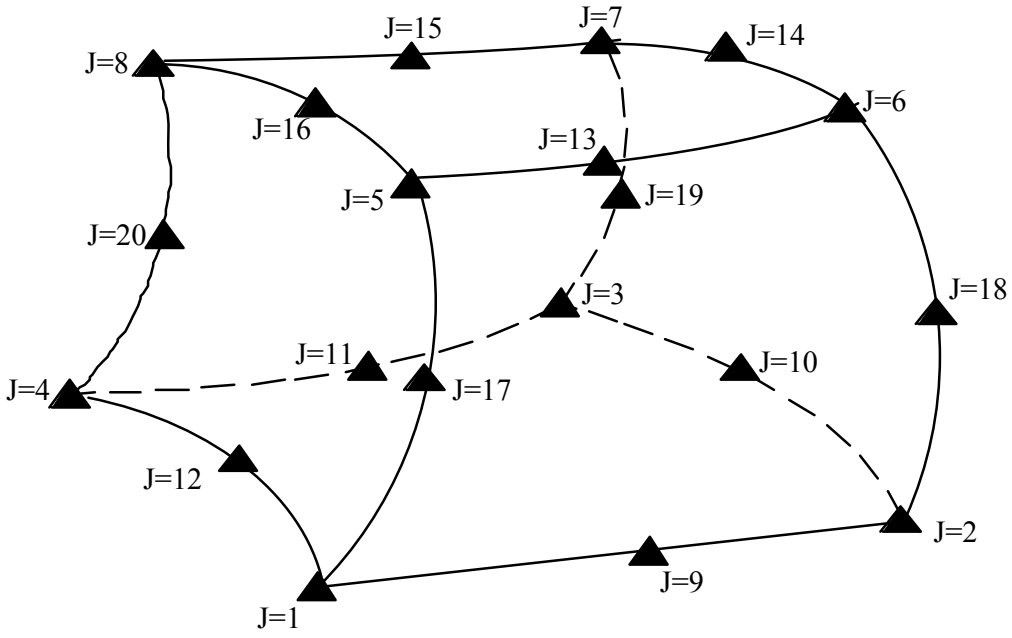


Figure 4.1.3 Nodal Generation Over a Surface



NUMGP = 8



NUMGP = 20

Figure 4.1.4 Nodal Generation Within a Volume

Notes . .

4.2 Cylindrical / Polar Coordinate Generation

4.2.1 Nodal Coordinate Data

Note	Variable	Default	Description
(1)	N	[0]	Node number ≥ 1 and ≤ NUMNP
(2)	NUMGP	[0]	Number of generation points ≥ 0 = 0, no generation > 0, generate data
(3)	R(N)	[0.0]	R-coordinate of node N
	□(N)	[0.0]	Θ -coordinate of node N (in degrees)
	Z(N)	[0.0]	Z-coordinate of node N

Notes/
(1) The coordinates of each node must be defined, but need not be read in order. If the coordinates of node N are input and/or generated more than one time, the last values take priority. *Terminate with a blank record.*

(2) If NUMGP is greater than zero, this record initiates an isoparametric data generation sequence. Records 2 to NUMGP of the sequence define the coordinates of the additional generation points (see Section 4.2.2). The final record of the sequence defines the nodal increment information (See Section 4.2.3). After the generation sequence is completed, additional nodal coordinate records, or generation sequences, may follow.

(3) The coordinates of node N are first computed as:

$$\underline{x}(1, N) = r \cos \theta$$
$$\underline{x}(2, N) = r \sin \theta$$
$$\underline{x}(3, N) = z$$

in the set of axes defined by the reference direction vectors: $[\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3]$. The coordinates are thereafter referred to the global coordinate axes via the following rotation and translation:

$$\mathbf{x} = \mathbf{x}_0 + \mathbf{R} \cdot \underline{\mathbf{x}}$$

where $\mathbf{R} = [\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3]$.

A. Generation Along a Line

The line may be defined by 2, or 3 generation points (see Figure 4.1.1), and the physical space may be 1, 2, or 3 dimensional.

In the case NUMGP = 2, linear interpolation takes place resulting in equally spaced nodal points along a line in the {r, θ, z} space.

In the case NUMGP = 3, quadratic interpolation is employed and graded nodal spacing along a line may be achieved by placing the third generation point (J = 3) off center. The spacing in this case may be determined from the following mapping:

$$\mathbf{X}_A = \mathbf{X}(\xi_A) = \frac{1}{2}\xi_A(\xi_A - 1)\mathbf{X}_1{}^g + \frac{1}{2}\xi_A(\xi_A + 1)\mathbf{X}_2{}^g + (1 - \xi_A)(1 + \xi_A)\mathbf{X}_3{}^g$$

where ξ_A is the location of node number A in ζ space (the nodes are placed at equal intervals in ζ-space); $\mathbf{X}_1{}^g$, $\mathbf{X}_2{}^g$ and $\mathbf{X}_3{}^g$ are the coordinates of the three generation points in {r, θ, z}-space; and $\mathbf{X}_A$ denotes the coordinates of the A^{th} node in {r, θ, z}-space (see Figure 4.1.2).

B. Generation Over a Surface

The surface may be defined by 4, or 8 generation points (see Figure 4.1.3) and the physical space may be 2, or 3 dimensional. In the 3-dimensional case, the surfaces may be curved.

In the case NUMGP = 4, bilinear interpolation is employed, resulting in equally spaced nodal points along generating lines.

In the case NUMGP = 8, biquadratic "serendipity" interpolation is employed and graded nodal spacing may be achieved by placing generation point 5-8 off center. Note that generation points 5-8 do not generally coincide with any nodal points. The spacing of the nodal points may be determined from the serendipity mapping.

C. Generation Over a Volume

The volume is brick shaped and may be defined by 8, or 20 generation points (see Figure 4.1.4). In this case the physical space must be 3-dimensional.

If NUMGP = 8, trilinear interpolation is employed, resulting in equally spaced nodal points along generating lines.

If NUMGP = 20, triquadratic serendipity interpolation is employed and graded nodal spacing may be achieved by placing generation points 9-20 off center. Note that generation points 9-20 do not generally coincide with any nodal points. The spacing of the nodal points may be determined by the serendipity mapping.

4.2.2 Generation Point Coordinate Data (NUMGP-1)

The coordinates of each generation point are defined by a generation point coordinate record. The records must be read in order (J = 2,..., NUMGP) following the nodal coordinate record which initiated the generation sequence (J = 1). A nodal record (see Section 4.2.3), which completes the sequence, must follow the last generation point record.

Note	Variable	Default	Description
	M	[0]	Node number
	MGEN	[0]	Generation parameter = 0, coordinates of the Jth generation point are input on this record; M is ignored = 1, coordinates of the J^{th} generation point are set equal to coordinates of node M which was previously defined; coordinates on this record are ignored.
	TEMP(R, J)	[0.0]	R-coordinate of generation point J
	TEMP(□, J)	[0.0]	□-coordinate of generation point J
	TEMP(Z, J)	[0.0]	Z-coordinate of generation point J

4.2.3 Nodal Increments Data

Note	Variable	Default	Description
	NINC(1)	[0]	Number of nodal increments in the R-direction
	INC(1)	[0]	Node number increment in the R-direction
	NINC(2)	[0]	Number of nodal increments in the Θ-direction
	INC(2)	[0]	Node number increment in the Θ-direction
	NINC(3)	[0]	Number of nodal increments in the Z-direction
	INC(3)	[0]	Node number increment in the Z-direction

Notes . .

4.3 Spherical Coordinate Input

4.3.1 Nodal Coordinate Data

Note	Variable	Default	Description
(1)	N	[0]	Node number ≥ 1 and ≤ NUMNP
(2)	NUMGP	[0]	Number of generation points ≥ 0 = 0, no generation > 0, generate data
(3)	R(N)	[0.0]	R-coordinate of node N
	Θ (N)	[0.0]	Θ -coordinate of node N (in degrees)
	Φ (N)	[0.0]	Φ -coordinate of node N (in degrees)

Notes/
(1) The coordinates of each node must be defined, but need not be read in order. If the coordinates of node N are input and/or generated more than one time, the last values take priority. *Terminate with a blank record.*

(2) If NUMGP is greater than zero, this record initiates an isoparametric data generation sequence. Records 2 to NUMGP of the sequence define the coordinates of the additional generation points (see Section 4.3.2). The final record of the sequence defines the nodal increment information (see Section 4.3.3). After the generation sequence is completed, additional nodal coordinate records, or generation sequences, may follow.

(3) The coordinates of node N are first computed as:

$$\underline{x}(1, N) = r \sin \phi \cos \theta$$
$$\underline{x}(2, N) = r \sin \phi \sin \theta$$
$$\underline{x}(3, N) = r \cos \phi$$

in the set of axes defined by the reference direction vectors: $[\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3]$. The coordinates are thereafter referred to the global coordinate axes via the following translation and rotation:

$$\mathbf{x} = \mathbf{x}_0 + \mathbf{R} \cdot \underline{\mathbf{x}}$$

where $\mathbf{R} = [\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3]$.

A. Generation Along a Line

The line may be defined by 2 or 3 generation points (see Figure 4.1.1), and the physical space may be 1, 2, or 3 dimensional.

In the case NUMGP = 2, linear interpolation takes place resulting in equally spaced nodal points along a line in the $\{r, \theta, \phi\}$ space.

In the case NUMGP = 3, quadratic interpolation is employed and graded nodal spacing along a line may be achieved by placing the third generation point (J = 3) off center. The spacing in this case may be determined from the following mapping:

$$\mathbf{X}_A = \mathbf{X}(\xi_A) = \frac{1}{2}\xi_A(\xi_A - 1)\mathbf{X}_1{}^g + \frac{1}{2}\xi_A(\xi_A + 1)\mathbf{X}_2{}^g + (1 - \xi_A)(1 + \xi_A)\mathbf{X}_3{}^g$$

where ξ_A is the location of node number A in ζ space (the nodes are placed at equal intervals in ζ-space); $\mathbf{X}_1{}^g$, $\mathbf{X}_2{}^g$ and $\mathbf{X}_3{}^g$ are the coordinates of the three generation points in $\{r, \theta, \phi\}$-space; and $\mathbf{X}_A$ denotes the coordinates of the A^{th} node in $\{r, \theta, \phi\}$-space (see Figure 4.1.2).

B. Generation Over a Surface

The surface may be defined by 4, or 8 generation points (see Figure 4.1.3) and the physical space may be 2, or 3 dimensional. In the 3-dimensional case, the surfaces may be curved.

In the case NUMGP = 4, bilinear interpolation is employed, resulting in equally spaced nodal points along generating lines.

In the case NUMGP = 8, biquadratic "serendipity" interpolation is employed and graded nodal spacing may be achieved by placing generation point 5-8 off center. Note that generation points 5-8 do not generally coincide with any nodal points. The spacing of the nodal points may be determined from the serendipity mapping.

C. Generation Over a Volume

The volume is brick shaped and may be defined by 8, or 20 generation points (see Figure 4.1.4). In this case the physical space must be 3-dimensional.

If NUMGP = 8, trilinear interpolation is employed, resulting in equally spaced nodal points along generating lines.

If NUMGP = 20, triquadratic serendipity interpolation is employed and graded nodal spacing may be achieved by placing generation points 9-20 off center. Note that generation points 9-20 do not generally coincide with any nodal points. The spacing of the nodal points may be determined by the serendipity mapping.

4.3.2 Generation Point Coordinate Data (NUMGP-1)

The coordinates of each generation point are defined by a generation point coordinate record. The records must be read in order (J = 2,..., NUMGP) following the nodal coordinate record which initiated the generation sequence (J = 1). A nodal record (see Section 4.3.3), which completes the sequence, must follow the last generation point record.

Note	Variable	Default	Description
	M	[0]	Node number
	MGEN	[0]	Generation parameter = 0, coordinates of the J^{th} generation point are input on this record; M is ignored = 1, coordinates of the J^{th} generation point are set equal to coordinates of node M which was previously defined; coordinates on this record are ignored
	TEMP(R, J)	[0.0]	R-coordinate of generation point J
	TEMP(□, J)	[0.0]	Θ -coordinate of generation point J
	TEMP(□, J)	[0.0]	Φ -coordinate of generation point J

4.3.3 Nodal Increments Data

Note	Variable	Default	Description
	NINC(1)	[0]	Number of nodal increments in the R-direction
	INC(1)	[0]	Node number increment in the R-direction
	NINC(2)	[0]	Number of nodal increments in the Θ -direction
	INC(2)	[0]	Node number increment in the Θ -direction
	NINC(3)	[0]	Number of nodal increments in the Φ -direction
	INC(3)	[0]	Node number increment in the Φ -direction

4.4 Lattice Coordinate Generation

The following information is used to generate the coordinates on a lattice system.

Note	Variable Name	Type	Default	Description
	nsd_lattice	integer	[0]	Number of spatial dimensions
	n1_lattice	integer	[1]	Number of points in direction 1
	n2_lattice	integer	[1]	Number of points in direction 2
	n3_lattice	integer	[1]	Number of points in direction 3
	dx1_lattice	real	[1.0]	lattice spacing in direction 1
	dx2_lattice	real	[1.0]	lattice spacing in direction 2
	dx3_lattice	real	[1.0]	lattice spacing in direction 3

5.0 NODAL RESTRAINT CONDITION DATA

5.1 Nodal Boundary Condition Data

NODAL_BOUNDARY_CONDITIONS

NODAL_BOUNDARY_CONDITIONS generation_type = type , etc...
node, ng, (id(i, node), i = 1, ndof)
< etc..., terminate with a blank record >

Boundary condition data must be input for each node which has one or more specified displacements, velocities or accelerations. Boundary condition codes for each node may be assigned the following values:

id(i, node) = 0 , unspecified (i.e., active degree of freedom)
id(i, node) = 1 , prescribed displacement
id(i, node) = 2 , prescribed velocity
id(i, node) = 3 , prescribed acceleration

where i = 1, 2,..., number_of_degrees_of_freedom. If prescribed, the value is assumed to be zero (0.0), unless it is assigned a nonzero value as described in Section 7.1.

Note	Variable Name	Type	Default	Description
	Generation_type **list** type_1 type_2 type_3	list	[*]	Generation case
	Variable **all** solid_displacement solid_rotation solid_displacement_and_rotation fluid_velocity temperature pressure potential electric_potential scalar_transport level_set stream_fct mesh_motion	list	[*]	Nodal unknowns selection: all degrees of freedom solid displacement solid rotation solid displacement and rotation fluid velocity temperature pressure potential electric potential scalar transport level set stream function ALE mesh motion
	File_name	string	[none]	File name (optional). Name must be enclosed in quotation marks.

• *No Generation (List) Case*

Nodal boundary conditions must follow in the form:
< Node_number , Restraint_X, Restraint_Y , Restraint_Z, etc...>
< etc..., terminate with a blank record >

• *Generation Case*

Nodal boundary conditions generation data must follow.
< terminate with a blank record >

EXAMPLE

```
Nodal_Boundary_Conditions /
   file_name = "boundary_file"          # Read displacements from file:  boundary_file
```

EXAMPLE

```
Nodal_Boundary_Conditions /
   generation_type = type_1             # Select generation option
   Nodal Boundary Condition Generation Data follow
```

5.1.1 Nodal Boundary Condition Generation Data Type 1

Note	Variable	Default	Description
(1)	NODE	[0]	Node number ≥ 1 and ≤ NUMNP
(2)	NG	[0]	Generation increment
(3)	ID(1, NODE)	[0]	Degree of freedom 1 boundary code
	ID(2, NODE)	[0]	Degree of freedom 2 boundary code
	.	.	.
	etc.	.	.
	.	.	.
	ID(NDOF, NODE)	[0]	Degree of freedom NDOF boundary code

Notes/
(1) Boundary condition data must be input for each node which has one or more specified displacements, velocities or accelerations. If more than one boundary condition data record for node N is input, the last one read takes priority. *Terminate with a blank record.*

(2) Boundary condition data can be generated by employing a two record sequence as follows:

Record 1: L, LG, ID(1, L),..., ID(NDOF, L)
Record 2: N, NG, ID(1, N),..., ID(NDOF, N)

The boundary codes of all nodes NODE:

L + LG, L + 2*LG,..., N - MOD(N-L, LG)

(i.e., less than N) are set equal to those of node L. If LG is blank or zero, no generation takes place between L and N.

(3) Boundary condition codes for each node may be assigned the following values:

id(i, node) = 0 , unspecified (i.e., active degree of freedom)
id(i, node) = 1 , prescribed displacement
id(i, node) = 2 , prescribed velocity
id(i, node) = 3 , prescribed acceleration

where i = 1, 2,..., NDOF. If prescribed, the value is assumed to be zero (0.0), unless it is assigned a nonzero value as described in Section 7.1.

5.1.2 Nodal Boundary Condition Generation Data Type 2

Note	Variable	Default	Description
(1)	NODE1	[0]	Node number 1 ≥ 1 and ≤ NUMNP
(1)	NODE2	[0]	Node number 2 ≥ 1 and ≤ NUMNP
(2)	NG	[0]	Generation increment
(3)	ID(1)	[0]	Degree of freedom 1 boundary code
	ID(2)	[0]	Degree of freedom 2 boundary code
	etc.	.	.
	ID(NDOF)	[0]	Degree of freedom NDOF boundary code

Notes/
(1) Boundary condition data must be input for each node which has one or more specified displacements, velocities or accelerations. If more than one boundary condition data for any node NODE is input, the last one read takes priority. *Terminate with a blank record.*

(2) If ng is non zero, boundary condition data for all nodes from node1 to node2 by increment ng can be generated. The boundary codes of all nodes:

NODE1 + NG, NODE1 + 2*NG,..., NODE2 - MOD(NODE2 - NODE1, NG)

(i.e., less than or equal to NODE2) are set equal to those of node NODE1. If either NODE2 or NG is zero, no generation takes place between NODE1 and NODE2.

(3) Boundary condition codes may be assigned the following values:

id(i, node) = 0 , unspecified (i.e., active degree of freedom)
id(i, node) = 1 , prescribed displacement
id(i, node) = 2 , prescribed velocity
id(i, node) = 3 , prescribed acceleration

where i = 1, 2,..., NDOF. If prescribed, the value is assumed to be zero (0.0), unless it is assigned a nonzero value as described in Section 7.1.

5.1.3 Nodal Boundary Condition Generation Data Type 3

Note	Variable	Default	Description
(1)	NODE	[0]	Node number ≥ 1 and ≤ NUMNP
(2)	NG	[0]	Generation code (=0 no generation, =1 generation)
(3)	ID(1, NODE)	[0]	Degree of freedom 1 boundary code
	ID(2, NODE)	[0]	Degree of freedom 2 boundary code
	.	.	.
	etc.	.	.
	.	.	.
	ID(NDOF, NODE)	[0]	Degree of freedom NDOF boundary code

Notes/
(1) Boundary condition data must be input for each node which has one or more specified displacements, velocities or accelerations. If more than one boundary condition data record for node N is input, the last one read takes priority. *Terminate with a blank record.*

(2) Boundary condition data can be generated by employing a two record sequence as follows:

Record 1: N, NG, ID(1, N),..., ID(NDOF, N)
Record 2: Nodal Increment Data

If NG is zero, no generation takes place.

(3) Boundary condition codes for each node may be assigned the following values:

id(i, node) = 0 , unspecified (i.e., active degree of freedom)
id(i, node) = 1 , prescribed displacement
id(i, node) = 2 , prescribed velocity
id(i, node) = 3 , prescribed acceleration

where i = 1, 2,..., NDOF. If prescribed, the value is assumed to be zero (0.0), unless it is assigned a nonzero value as described in Section 7.1

5.1.3.1 Nodal Increments Data

Note	Variable	Default	Description
	NINC(1)	[0]	Number of nodal increments for direction 1
	INC(1)	[0]	Node number increment for direction 1
(1)	NINC(2)	[0]	Number of nodal increments for direction 2
	INC(2)	[0]	Node number increment for direction 2
(1)	NINC(3)	[0]	Number of nodal increments for direction 3
	INC(3)	[0]	Node number increment for direction 3

Notes/
(1) Each option is assigned an option code (IOPT) as follows:

IOPT	*Option*
1	Generation along a line
2	Generation over a surface
3	Generation over a volume

IOPT is determined by the following logic:

```
IOPT = 3
IF(NINC(3) = 0)   IOPT = 2
IF(NINC(2) = 0)   IOPT = 1
```

5.2 Slaved Nodes

SLAVE_NODES

SLAVE_NODES file_name = "<string>"
node1, node2, ng , (idof1(i), idof2(i), i = 1, ndof)
< etc..., terminate with a blank record >

Nodes may be slaved to share the same equation number for any selected degree of freedom. Such an option is useful, i.e., for modeling cyclic symmetry in structures, for modeling free-field conditions in seismic calculations, etc...

Note	Variable Name	Type	Default	Description
	Input_format keywords / **list**	list	[*]	Select input format option
	File_name	string	[none]	File name (optional). Name must be enclosed in quotation marks.

5.2.1 **Slaved Nodes Data** (Keywords Read Method)

Note	Variable Name	Type	Default	Description
(1)	NODE1	integer	[0]	Node number 1
	NODE2	integer	[0]	Node number 2
(2)	NG	integer	[0]	Generation increment
(3)	Variable_i_node1	string	[none]	Variable_i at NODEl
(3)	Variable_i_node2	string	[none]	Variable_i at NODE2
	. etc.			

Notes/
(1) Node numbers NODE1 and NODE2（NODE 2 ≥ NODE1）are assigned to share the same equation numbers for the degrees of freedom listed in variable_i_node1 and variable_i_node2. *Terminate with a blank record.*

(2) See note 2 in Section 5.2.2

(3) Variable_i_node1 and variable_i_node2 are assigned by using the following list of names:

solid_motion_i	(i = 1, Nsd)
solid_rotation_i	(i = 1, Nsd)
fluid_motion_i	(i = 1, Nsd)
pressure_i	(i = 1, Number_of_phases)
temperature	
potential	
scalar_transport_i	(i = 1, Number_of_components)
mesh_motion_i	(i = 1, Nsd)

5.2.2 **Slaved Nodes Data** (List Read Method)

Note	Variable	Default	Description
(1)	NODE1	[0]	Node number 1
	NODE2	[0]	Node number 2
(2)	NG	[0]	Generation increment
(3)	IDOF1(I)	[0]	Degree of freedom number for NODE1
	IDOF2(I)	[0]	Degree of freedom number for NODE2
	etc.	.	.
	IDOF1(NDOF)	[0]	Degree of freedom number for NODE1
	IDOF2(NDOF)	[0]	Degree of freedom number for NODE2

Notes/
(1) Node numbers NODE1 and NODE2（NODE 2 ≥ NODE 1）are assigned to share the same equation numbers for the degrees of freedom listed in IDOF1(I) and IDOF2(I). *Terminate with a blank record.*

(2) Slaved condition data can be generated by employing a two record sequence as follows:

Record 1: LODE1, LODE2, LG, (IDOF1(I), IDOF2(I), I = 1, NDOF)
Record 2: NODE1, NODE2, NG, (IDOF1(I), IDOF2(I), I = 1, NDOF)

The slaved conditions codes of all nodes:

LODE1 + LG, LODE1 + 2*LG,..., NODE1 - MOD(NODE1 - LODE1, LG)
LODE2 + LG, LODE2 + 2*LG,..., NODE2 - MOD(NODE2 - LODE2, LG)

(i.e., less than NODE1 and NODE2) are set equal to those of node LODE1 and LODE2. If LG is blank or zero, no generation takes place.

5.3 Equivalence Nodes

EQUIVALENCE_NODES

EQUIVALENCE_NODES file_name = "<string>"
node1, node2, ng
< etc..., terminate with a blank record >

Nodes may be equivalenced to share the same equation number for all degrees of freedom.

5.3.1 Equivalence Nodes Data

Note	Variable	Default	Description
(1)	NODE1	[0]	Node number 1
	NODE2	[0]	Node number 2
(2)	NG	[0]	Generation increment

Notes/
(1) Node numbers NODE1 and NODE2（NODE 2 ≥ NODE1）are assigned to share the same equation numbers for all degrees of freedom. *Terminate with a blank record.*

(2) Equivalence nodes can be generated by employing a two record sequence as follows:

Record 1: LODE1, LODE2, LG
Record 2: NODE1, NODE2, NG

All nodes:

LODE1 + LG, LODE1 + 2*LG,..., NODE1 - MOD(NODE1 - LODE1, LG)
LODE2 + LG, LODE2 + 2*LG,..., NODE2 - MOD(NODE2 - LODE2, LG)

(i.e., less than NODE1 and NODE2) are set to be equivalenced. If LG is blank or zero, no generation takes place.

5.4 Euler-Lagrange Nodes

ALE_NODES

ALE_NODES file_name = "<string>"
node, ng, (id(i, node), i = 1, NSD)
< etc..., terminate with a blank record >

Nodal degrees of freedom may be selectively assigned to be either Lagrangian or of free surface degrees of freedom (see Section 9.2.0.9). This option is used in arbitrary Lagrangian Eulerian (ALE) computations.

5.4.1 Euler-Lagrange Nodal Data

Note	Variable	Default	Description
(1)	NODE	[0]	Node number ≥ 1 and ≤ NUMNP
(2)	NG	[0]	Generation increment
(3)	ID(1,NODE)	[0]	Degree of freedom 1 Euler-Lagrange code
	ID(2,NODE)	[0]	Degree of freedom 2 Euler-Lagrange code
	.	.	.
	etc.	.	.
	.	.	.
	ID(NSD,NODE)	[0]	Degree of freedom NSD Euler-Lagrange code

Notes/
(1) Euler-Lagrange condition data must be input for each node which is not ALE in one or more directions. Records need not be input in order. *Terminate with a blank record.*

(2) Euler-Lagrange nodal condition data can be generated by employing a two record sequence as follows:

Record 1: L, LG, ID(1, L),..., ID(NSD, L)
Record 2: N, NG, ID(1, N),..., ID(NSD, N)

The Euler-Lagrange codes of all nodes NODE:

L + LG, L + 2*LG,..., N - MOD(N-L, LG)

(i.e., less than N) are set equal to those of node L. If LG is blank or zero, no generation takes place between L and N.

(3) Euler-Lagrange condition codes may be assigned the following values:

ID(I, NODE) = 0 , ALE degree of freedom
ID(I, NODE) = 1 , Lagrange degree of freedom
ID(I, NODE) = 2 , free surface degree of freedom

where I = 1, 2,..., NSD. If more than one Euler-Lagrange condition data record for node NODE is input, the last one read in takes priority

Notes . .

Notes . .

6.0 INITIAL NODAL KINEMATIC CONDITION DATA

6.1 Nodal Initial Displacement Data

INITIAL_D0

INITIAL_D0 generation_type = type , etc...
n, numgp, (d(i, n), i = 1 , ndof)
< etc..., terminate with a blank record >

Create the list of initial displacements. Two options are available. The initial displacements may be read in directly as a list (optionally from another file), or may be generated.

Note	Variable Name	Type	Default	Description
	Generation_type **list** Cartesian	list	[*]	Generation case
	Variable **all** solid_displacement solid_displacement_and_rotation fluid_velocity temperature pressure potential electric_potential scalar_transport level_set stream_fct solid_displacement_and_fluid_velocity solid_displacement_and_fluid_pressure solid_displacement_and_temperature mesh_motion	list	[*]	Nodal unknowns selection: all degrees of freedom solid displacement solid displacement and rotation fluid velocity temperature pressure potential electric potential scalar transport level set stream function porous continuum porous continuum thermo-solid continuum ALE mesh motion
	File_name	string	[none]	File name (optional). Name must be enclosed in quotation marks.

• *No Generation (List) Case*

Initial nodal displacements must follow in the form:
< Node_number, Displacement_X, Displacement_Y, Displacement_Z, etc...>
< etc..., terminate with a blank record >

• *Generation Case*

Initial Nodal Displacements Generation Data must follow.
< terminate with a blank record >

```
EXAMPLE
      Initial_D0  /
        file_name = "initial_d0_file"              # read displacements from file:  initial_d0_file

EXAMPLE
      Initial_D0  /
        generation_type = cartesian                # Select generation option
        Initial Nodal Displacements Generation Data follow
```

6.1.1 Nodal Initial Displacement Generation Data

Note	Variable	Default	Description
(1)	N	[0]	Node number ≥ 1 and ≤ NUMNP
(2)	NUMGP	[0]	Number of generation points = 0, no generation ≠ 0, generate data
(3)	D(1, N)	[0.0]	Degree of freedom 1 initial displacement
	D(2, N)	[0.0]	Degree of freedom 2 initial displacement
	.	.	.
	etc.	.	.
	.	.	.
	D(NDOF, N)	[0.0]	Degree of freedom NDOF initial displacement

Notes/
(1) Initial displacement data must be included for each node subjected to non-zero initial displacement. Terminate with a blank record.

(2) If NUMGP is greater than zero, this records initiates an isoparametric data generation sequence. The scheme used is the same as the one used for coordinate generation described previously in Chapter 4. Records 2 to NUMGP of the sequence define the initial displacements of the additional generation points (see Section 6.2). The final record of the sequence defines the nodal increment information, and is identical to the one used for coordinate generation (see Section 4.3). After the generation sequence is completed, additional nodal initial displacement records, or generation sequences, may follow.

The generation may be performed along a line, over a surface, or over a volume. A description of these options is given below.

A. Generation Along a Line

Generation of data along a line may be performed using 2, or 3 generation points (see Figure 6.3), and the physical space (**D**-space) may be 1, 2, or 3 dimensional.

If NUMGP = 2, linear interpolation takes place with respect to ξ-space. If the values are equally spaced in **D**-space, then the variation will also be linear in **D**-space. Otherwise a nonlinear variation will be induced by the unequal values spacing.

If NUMGP = 3, quadratic interpolation is performed with respect to space. Note that the third generation point does not generally coincide with a nodal point. The variation of data may be determined from the following mapping:

$$\mathbf{D}_A = \mathbf{D}(\xi_A) = \frac{1}{2}\xi_A(\xi_A - 1)\mathbf{D}_1{}^g + \frac{1}{2}\xi_A(\xi_A + 1)\mathbf{D}_2{}^g + (1 - \xi_A)(1 + \xi_A)\mathbf{D}_3{}^g$$

where ξ_A is the location of node number A in ξ-space (recall that the nodes are assumed to be placed at equal intervals in ξ-space); $\mathbf{D}_1{}^g$, $\mathbf{D}_2{}^g$ and $\mathbf{D}_3{}^g$ are the displacement data assigned to the three generation points (i.e., ξ = -1, +1, and 0, resp.); and $\mathbf{D}_A$ is the generated displacement data at node A.

The case in which NUMGP = 2, may be deduced from the case NUMGP = 3 by setting $\mathbf{D}_3{}^g = (\mathbf{D}_1{}^g + \mathbf{D}_2{}^g)/2$

B. Generation Over a Surface

Generation of data may be performed using 4, or 8 generation points. The generation points and nodal patterns are the same as in coordinate generation (see Figure 4.1.3).

In the case NUMGP = 4, bilinear interpolation is performed; for NUMGP = 8, biquadratic "serendipity" interpolation is performed. Note that generation points 5-8 do not in general coincide with nodal points.

C. Generation Over a Volume

Generation of data over a brick-shaped volume may be performed using 8, or 20 generation points. The generation points and nodal patterns are the same as for coordinate generation (see Figure 4.1.4).

If NUMGP = 8, trilinear interpolation is employed; if NUMGP = 20, triquadratic "serendipity" interpolation is employed. Note that generation points 9-20 do not in general coincide with nodal points.

(3) The elements of the array D(NDOF,NUMNP) are initialized to zero. If the initial displacements of node N are input and/or generated more than one time, the last values take priority.

6.1.2 Generation Point Initial Displacement Data (NUMGP-1)

The initial displacements of each generation point are defined by a generation point initial displacement record. The records must be read in order (J = 2, 3,...,NUMGP) following the nodal initial displacement record which initiated the generation sequence (J = 1). A nodal increments record (see Section 6.1.3), which completes the sequence, must follow the last generation point record (J = NUMGP).

Note	Variable	Default	Description
	M	[0]	Node number
	MGEN	[0]	Generation parameter = 0, initial displacements of the J^{th} generation point are input on this card; M is ignored = 1, initial displacements of the J^{th} generation point are set equal to initial displacements of node M which was previously defined; initial displacements on this card are ignored
	TEMP(1,J)	[0.0]	Degree of freedom 1 initial displacement of generation point J
	TEMP(2,J)	[0.0]	Degree of freedom 2 initial displacement of generation point J
	.	.	.
	etc.	.	.
	.	.	.
	TEMP(NDOF,J)	[0.0]	Degree of freedom NDOF initial displacement of generation point J

6.1.3 Nodal Increments Data

Note	Variable	Default	Description
	NINC(1)	[0]	Number of nodal increments for direction 1
	INC(1)	[0]	Node number increment for direction 1
(1)	NINC(2)	[0]	Number of nodal increments for direction 2
	INC(2)	[0]	Node number increment for increment 2
(1)	NINC(3)	[0]	Number of nodal increments for direction 3
	INC(3)	[0]	Node number increment for direction 3

Notes/
(1) Each option is assigned an option code (IOPT) as follows:

IOPT	*Option*
1	Generation along a line
2	Generation over a surface
3	Generation over a volume

IOPT is determined by the following logic:

```
IOPT = 3
IF(NINC(3) = 0)   IOPT = 2
IF(NINC(2) = 0)   IOPT = 1
```

6.2 Nodal Initial Velocity Data

INITIAL_V0

INITIAL_V0 generation_type = type , etc...
n, numgp, (v(i, n), i = 1 , ndof)
< etc..., terminate with a blank record >

Create the list of initial nodal velocities. Two options are available: the initial velocities may be read in directly as a list (optionally from another file), or may be generated. Use same sequence as for nodal initial displacement generation data.

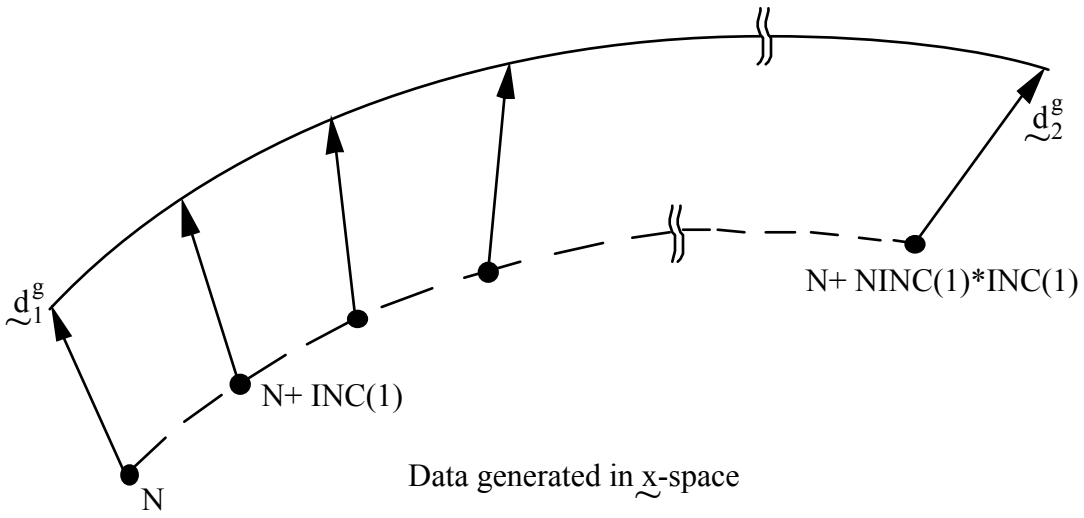
Notes . .

6.3 Background Nodal Field Data

BACKGROUND_NODAL_FIELD

BACKGROUND_NODAL_FIELD generation_type = type , etc...
n, numgp, (v(i, n), i = 1 , ncomp)
< etc..., terminate with a blank record >

Create a background field of nodal velocities. Two options are available: the velocities may be read in directly as a list (optionally from another file), or may be generated. Use same sequence as for nodal initial displacement generation data.



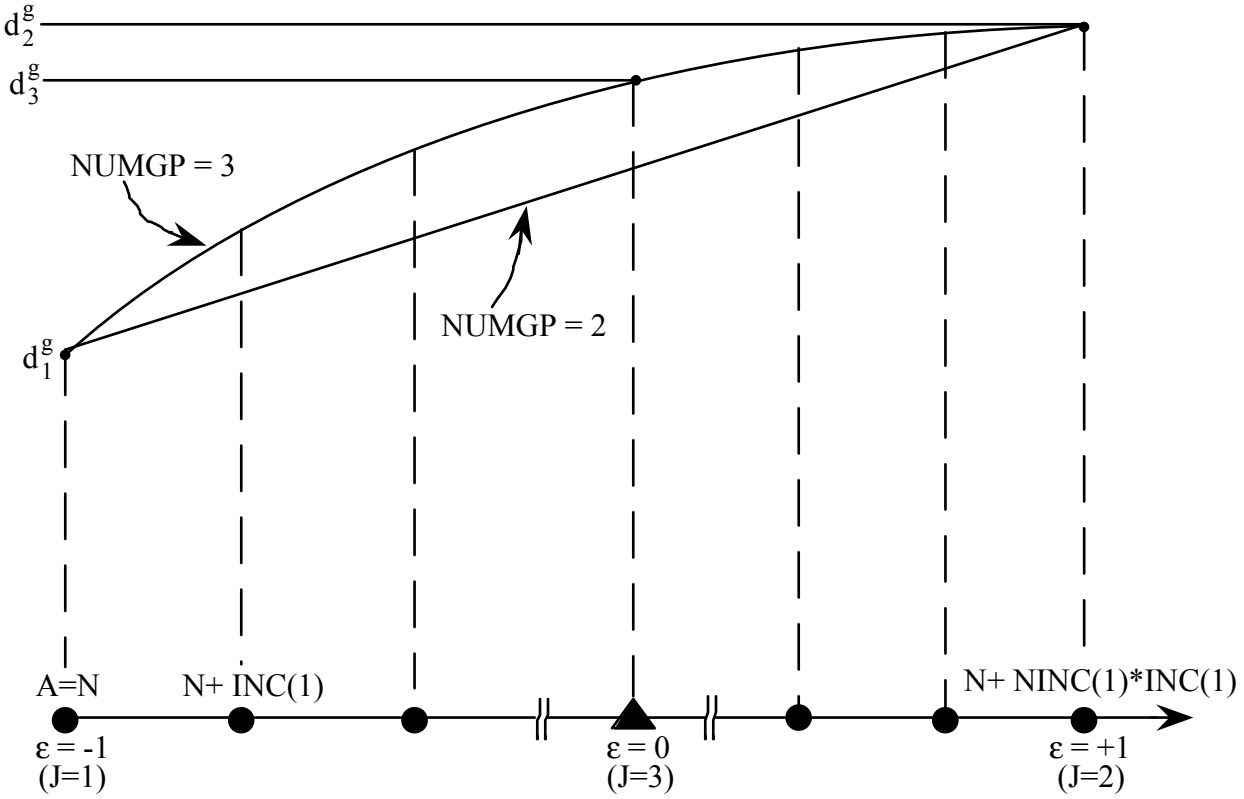


Figure 6.3 Displacement / Velocity Vector Generation

7.0 PRESCRIBED BOUNDARY VALUE DATA

7.1 Nodal Forces and Kinematics Data

Applied nodal forces and kinematics are defined by an expansion of the form:

$$\mathbf{F}(\mathbf{X}, t) = \sum_{i=1,NLC} \mathbf{F}(\mathbf{X}, i) * G[j, t] \text{ with } j = \text{load\_time\_function (i)}, \ i = \text{Load\_case}$$

where **F**(**X**, t) is the resultant force or kinematics acting at node A with coordinate **X** at time t; G is the load time function of the ith load case; F is the "mode shape" of the ith load case; and NLC is the total number of nodal load cases defined on the first control record (see Section 2.1). The data preparation for the load-time functions is described in Section 8.0. In this section, the data preparation for the F's is described. The value specified is a force if the corresponding nodal boundary condition restraint code (see Section 5.1) is zero, or a displacement/velocity/acceleration if the corresponding restraint code is non-zero (see Section 5.1).

NODAL_LOADS

NODAL_LOADS load_time_function_number = ltime , etc...
< etc..., terminate with a blank record >

Create the list of nodal loads. The value specified is a force if the corresponding nodal boundary condition restraint code is zero, or a displacement, velocity or acceleration if the corresponding restraint code is non-zero (see Section 5.1). Two options are available. The nodal loads may be read in directly as a list (optionally from another file), or may be generated.

Note	Variable Name	Type	Default	Description
	Load_case_number	integer	[*]	Load case number (only required in Restart mode)
	Generation_type **list** Cartesian	list	[*]	Generation case
	Variable **all** solid_displacement solid_rotation solid_displacement_and_rotation fluid_velocity temperature pressure potential electric_potential scalar_transport level_set stream_fct mesh_motion solid_force bending_moment fluid_traction pressure_gradient heat_flux potential_flux electric_potential_flux scalar_flux	list	[*]	Nodal variables selection: all degrees of freedom solid displacement solid rotation solid displacement and rotation fluid velocity temperature pressure potential electric potential scalar transport level set stream function ale mesh motion solid force bending moment fluid traction pressure gradient heat flux potential flux electric potential flux scalar transport flux
	Type **constant** / time_dependent	list	[*]	Load type
	Space_time_dependence **uniform** / nonuniform	list	[*]	Space time dependence

(cont'd)

(cont'd)

Note	Variable Name	Type	Default	Description
	Load_time_function_number	integer	[0]	Load time function number (only applicable to uniform space time dependence case)
	File_name	string	[none]	File name (optional). Name must be enclosed in quotation marks

• *No Generation (List) Case*

Nodal loads must follow in the form:

– Uniform space time dependence case:

< Node_number , load_X , load_Y , load_Z , etc...>
< etc..., terminate with a blank record >.

– Nonuniform space time dependence case:

< Node_number , load_time_function_number, load_X , load_Y , load_Z , etc...>
< etc..., terminate with a blank record >.

• *Generation Case* (Only applicable to uniform space time dependence.)

Nodal loads generation data must follow.
< terminate with a blank record >.

EXAMPLE

```
NODAL_LOADS  \
  load_time_function_number = ltime \
  file_name = "nodal_load_file"          # read nodal loads from file: nodal_load_file
```

7.1.1 Applied Nodal Forces and/or Kinematics Generation Data

Note	Variable	Default	Description
(1)	N	[0]	Node number ≥ 1 and ≤ NUMNP
(2)	NUMGP	[0]	Number of generation points ≥ 0 = 0, no generation ≠ 0, generate data
(3)	F(1,N,LC)	[0.0]	Degree of freedom 1 force or kinematics
	F(2,N,LC)	[0.0]	Degree of freedom 2 force or kinematics
	-	-	-
	etc.	-	etc.
	-	-	-
	F(NDOF,N,LC)	[0.0]	Degree of freedom NDOF force or kinematics

Notes/
(1) Applied nodal force or kinematics data must be included for each node subjected to a nonzero applied force or nonzero prescribed kinematics. *Terminate with a blank record.*

(2) If NUMGP is greater than zero, this record initiates an isoparametric data generation sequence. The scheme used is the same as the one for coordinate and initial displacement/velocity generation (see Chapters 4 and 6, respectively). Records 2 to NUMGP of the sequence define the applied forces/kinematics of the additional generation points (see Section 7.2). The final records of the sequence defines the nodal increment information and is identical to the one used for coordinate generation (see Section 4.3). After the generation sequence is completed, additional nodal applied forces/kinematics data records, or generation sequences, may follow.

The generation may be performed along a line, over a surface, or over a volume. For additional information concerning these options see Note (2) of Section 6.1.

(3) The elements of the array F(NDOF,NUMNP,NLC) are initialized to zero. If the applied forces/kinematics of node N are input and/or generated more than one time, the last value takes priority.

7.1.2 Generation Applied Nodal Forces / Kinematics Data

The applied forces/kinematics of each generation point are defined by a generation point applied force/kinematics record. The records must be read in order (J = 2, 3,...,NUMGP) following the nodal applied force/prescribed kinematics record which initiated the generation sequence (J = 1). A nodal increments record (see Section 7.1.3) follows the last generation point record (J = NUMGP) and completes the sequence.

Note	Variable	Default	Description
	M	[0]	Node number
	MGEN	[0]	Generation parameter = 0, applied forces/kinematics of the J^{th} generation points are input on this record; M is ignored = 1, applied forces/kinematics of the J^{th} generation point are set equal to applied forces/kinematics of node M which were previously defined; applied forces/ kinematics on this record are ignored.
	TEMP(1,J)	[0.0]	Degree of freedom 1 applied force/kinematics for generation point J
	TEMP(2,J)	[0.0]	Degree of freedom 2 applied force/kinematics for generation point J
	-	-	-
	etc.	-	etc.
	-	-	-
	TEMP(NDOF,J)	[0.0]	Degree of freedom NDOF applied force/kinematics for generation point J

7.1.3 Nodal Increments Data

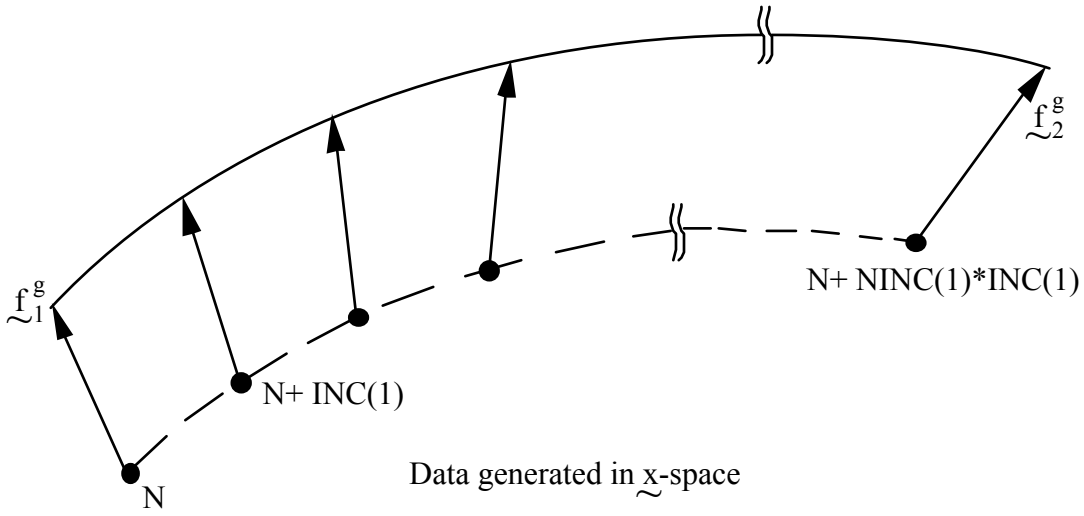
Note	Variable	Default	Description
	NINC(1)	[0]	Number of nodal increments for direction 1
	INC(1)	[0]	Node number increment for direction 1
(1)	NINC(2)	[0]	Number of nodal increments for direction 2
	INC(2)	[0]	Node number increment for increment 2
(1)	NINC(3)	[0]	Number of nodal increments for direction 3
	INC(3)	[0]	Node number increment for direction 3

Notes/
(1) Each option is assigned an option code (IOPT) as follows:

IOPT	*Option*
1	Generation along a line
2	Generation over a surface
3	Generation over a volume

IOPT is determined by the following logic:

```
IOPT = 3
IF(NINC(3) = 0)   IOPT = 2
IF(NINC(2) = 0)   IOPT = 1
```



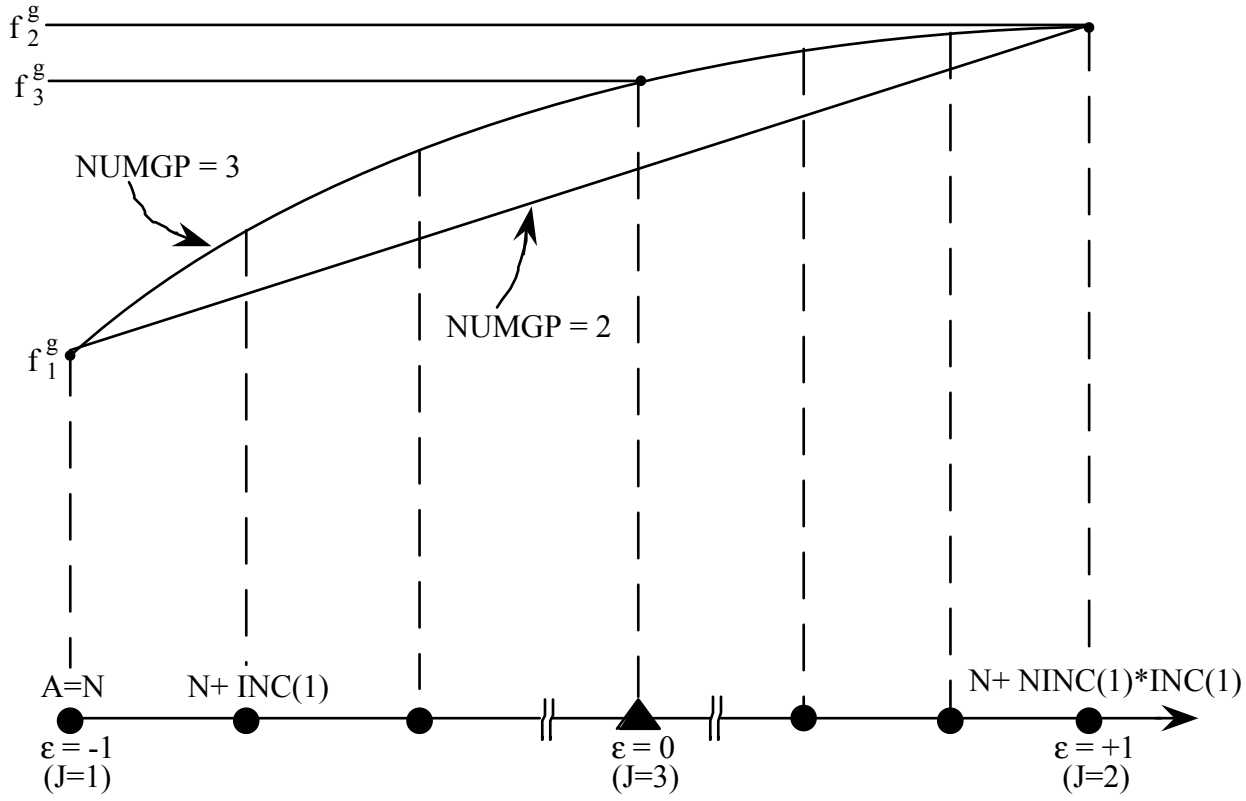


Figure 7.1.3 Load Vector Generation

Notes . .

7.2 Surface Loads

SURFACE_LOADS

SURFACE_LOADS load_time_function_number = ltime, etc...
n, (pres(j, n) , j = 1 , ncomp) < for n = 1 , nload >
< connectivity data > (see Chapter 11)

Note	Variable Name	Type	Default	Description
	Load_case_number	integer	[*]	Load case number (only required in Restart mode)
	Generation_type **list** Cartesian	list	[list]	Generation type
	Type **constant** / time_dependent	list	[*]	Load type
	Load_time_function_number	integer	[0]	Load time function number
	Number_of_loads	integer	[0]	Number of surface loads: Nload ≥ 0 (only required for no generation case).
	Load_direction **normal** / x_1 / x_2 / x_3	list	[normal]	Load direction normal = normal to element face/edge x_1 = in x_1 direction x_2 = in x_2 direction x_3 = in x_3 direction
(1)	Phase_number	integer	[1]	Phase number
	Element_shape four_node_quad eight_node_quad three_node_tri six_node_tri **two_node_line** three_node_line	list	[*]	Element shape
	Geometry_type **plane** axisymmetric three_dimensional	list	[plane]	Geometry option
	Finite_deformation on / **off**	list	[off]	Finite Deformation option If on: use Updated (Deformed) Geometry
	File_name	string	[none]	File name (optional). Name must be enclosed in quotation marks.

Note/
(1) Only applicable to porous media.

7.2.1 No Generation (List) Case

7.2.1a Surface Loads (Nload sets) – Two-Dimensional Case: NEN = 2, 3

Note	Variable	Default	Description
(1)	N	[0]	Load set number ≥ 1 and ≤ Nload
(2)	PRES1(1, N)	[0.0]	Normal traction at node 1
	-	-	-
	etc.	-	etc.
	-	-	-
	PRES1(Nen, N)	[0.0]	Normal traction at node Nen
(3)	SHEAR(1, N)	[0.0]	Shear stress at node 1
	-	-	-
	etc.	-	etc.
	-	-	-
	SHEAR(Nen, N)	[0.0]	Shear stress at node Nen

Notes/
(1) Each surface load must be input on a surface load record. The total number of surface loads must equal NLOAD read in on the control record. The records need not be read in any particular order.

(2) The normal traction / pressure (force / area) is assumed to be positive pointing inward, and is linearly interpolated between nodal values. The positive outward normal direction is obtained by rotating by - 90 degrees the direction (node 1 to node 2).

(3) The shear stress is assumed to be positive pointing in the direction (node 1 to node 2), and is linearly interpolated between nodal values.

7.2.1b Surface Loads (Nload sets) – Three-Dimensional Case: NEN = 3, 4, 6, 8

Note	Variable	Default	Description
(1)	N	[0]	Load set number ≥ 1 and ≤ Nload
(2)	PRES1(1, N)	[0.0]	Normal traction at node 1
	-	-	-
	etc.	-	etc.
	-	-	-
	PRES1(Nen, N)	[0.0]	Normal traction at node Nen

Notes/
(1) Each surface load must be input on a surface load record. The total number of surface loads must equal NLOAD read in on the control record. The records need not be read in any particular order.

(2) The normal traction / pressure (force / area) is assumed to be positive pointing inward, and is linearly interpolated between nodal values. The positive outward normal direction is obtained as the vector product of the directions (node 1 to node 2) X (node 1 to node 4).

7.2.2 Applied Surface Loads Generation Data

7.2.2.1 Applied Surface Loads Generation Data

Note	Variable	Default	Description
(1)	N	[0]	Node number ≥ 1 and ≤ NUMNP
(2)	NUMGP	[0]	Number of generation points ≥ 0 = 0, no generation ≠ 0, generate data
	F(1,N)	[0.0]	Normal traction at node N
	F(2,N)	[0.0]	Tangential traction at node N
	-	-	-

Notes/
(1) Applied surface loads data must be included for each node subjected to a nonzero applied surface loads. *Terminate with a blank record.*

(2) If NUMGP is greater than zero, this record initiates an isoparametric data generation sequence. The scheme used is the same as the one for coordinate and initial displacement/velocity generation (see Chapters 4 and 6, respectively). Records 2 to NUMGP of the sequence define the applied surface loads of the additional generation points (see Section 7.2). The final records of the sequence defines the nodal increment information and is identical to the one used for coordinate generation (see Section 4.3). After the generation sequence is completed, additional applied surface loads data records, or generation sequences, may follow.

The generation may be performed along a line, over a surface, or over a volume. For additional information concerning these options see Note (2) of Section 6.1.

7.2.2.2 Generation Surface Loads Data

The applied surface loads of each generation point are defined by a generation point surface loads record. The records must be read in order (J = 2, 3,...,NUMGP) following the applied surface loads record which initiated the generation sequence (J = 1). A nodal increments record (see Section 7.1.3) follows the last generation point record (J = NUMGP) and completes the sequence.

Note	Variable	Default	Description
	M	[0]	Node number
	MGEN	[0]	Generation parameter = 0, applied surface loads of the J^{th} generation points are input on this record; M is ignored = 1, applied surface loads of the J^{th} generation point are set equal to applied surface loads of node M which were previously defined; applied surface loads on this record are ignored.
	TEMP(1,J)	[0.0]	Normal traction at generation point J
	TEMP(2,J)	[0.0]	Tangential traction at generation point J
	-	-	-

7.2.2.3 Nodal Increments Data

Note	Variable	Default	Description
	NINC(1)	[0]	Number of nodal increments for direction 1
	INC(1)	[0]	Node number increment for direction 1
(1)	NINC(2)	[0]	Number of nodal increments for direction 2
	INC(2)	[0]	Node number increment for increment 2
(1)	NINC(3)	[0]	Number of nodal increments for direction 3
	INC(3)	[0]	Node number increment for direction 3

Notes/
(1) Each option is assigned an option code (IOPT) as follows:

IOPT	*Option*
1	Generation along a line
2	Generation over a surface
3	Generation over a volume

IOPT is determined by the following logic:

```
IOPT = 3
IF(NINC(3) = 0)   IOPT = 2
IF(NINC(2) = 0)   IOPT = 1
```

7.2.3 Surface Nodal Connectivity Data

Consult Chapter 11 for details. For 2D problems NEN = 2 or 3, and for 3D problems NEN = 3, 4, 6 or 8.

Notes . .

7.3 Convective / Radiative Surfaces

CONVECTIVE_SURFACE

CONVECTIVE_SURFACE number_of_loads = nload , etc...
m , hx(m) , Teta(m) , n(m) ,
T_0(m) < for m = 1 , number_of_loads >
< connectivity data > (see Chapter 11)

Note	Variable Name	Type	Default	Description
	Load_case_number	integer	[*]	Load case number (only required in Restart mode)
	Number_of_loads	integer	[1]	Number of Surface Loads: Nload ≥ 1
	Element_shape four_node_quad eight_node_quad three_node_tri six_node_tri **two_node_line** three_node_line	list	[*]	Element shape
	Geometry_type one_dimensional **plane** axisymmetric three_dimensional	list	[*]	Geometry Option
	Finite_deformation on / **off**	list	[off]	Finite Deformation Option If on: Use Updated (Deformed) Geometry
	File_name	string	[none]	File name (optional). Name must be enclosed in quotation marks.

7.3.1 Surface Loads Data

Note	Variable	Default	Description
	M	[0]	Load set number ≥ 1 and ≤ Nload
(1)	h_x(M)	[0.0]	Heat transfer coefficient
	Teta(M)	[0.0]	Surrounding (absolute) temperature
(2)	n(M)	[0.0]	Power exponent
	T_0(M)	[273.15]	Reference temperature

Notes/
(1) The heat flux (per unit area) generated by each surface is computed as:

$$q = h_x (T^n - Teta^n)$$

where T = (□ + T_0) = (absolute) temperature of solid medium. For radiative boundaries, the heat transfer coefficient hx is defined as:

$$h_x = \sigma\varepsilon$$

where:

σ = Stefan-Boltzman constant = 5.6697 10^{-8} Watt / m^2 / (°K)^4
ε = surface emissivity

(2) For Convective boundaries: n = 1; for Radiative boundaries: n = 4.

7.3.2 Surface Nodal Connectivity Data

Consult Chapter 11 for details. For 1D problems NEN = 1, for 2D problems NEN = 2 or 3, and for 3D problems NEN = 3, 4, 6 or 8.

7.4 Mixed Euler-Lagrange Free Surface

ALE_FREE_SURFACE

ALE_FREE_SURFACE element_shape = etc...
< connectivity data > (see Chapter 11)

Note	Variable Name	Type	Default	Description
	Element_shape four_node_quad two_node_line	list	[*]	Element shape
	Geometry_type plane axisymmetric three-dimensional	list	[*]	Geometry option

7.4.1 Free Surface Nodal Connectivity Data

Consult Chapter 11 for details. For 2D problems NEN=2, and for 3D problems NEN=4.

Notes . .

7.5 Free-Field Motion Options

7.5.1 Write Free-Field Motion Request

WRITE_MOTION

WRITE_MOTION	file_name = "<string>"

Note	Variable Name	Default	Description
(1)	File_name	[none]	Name of file to contain motion. Name must be enclosed in quotation marks.

Notes/
(1) This option allows the computed motion of all nodes to be written on a file.

7.5.2 Read Free-Field Motion Request

READ_FREE_FIELD_MOTION

READ_FREE_FIELD_MOTION	file_name = "<string>"

Note	Variable Name	Default	Description
(1)	File_name	[none]	Name of file which contains motion. Name must be enclosed in quotation marks.

Notes/
(1) This option allows the motion to be read from a file.

7.5.3 Filter Free-Field Motion Request

FILTER_FREE_FIELD_MOTION

	FILTER_FREE_FIELD_MOTION	start_node_number = start, etc...

This allows the motions of nodes in the free-field to be filtered to remove rigid body motions induced by liquefaction in the free-field. The motions of all nodes: starting node to ending node by node generation increment, relative to the motions of the reference node number, are then filtered.

Note	Variable Name	Type	Default	Description
	Start_node_number	integer	[0]	Starting node number > 0
	End_node_number	integer	[0]	Ending node number ≥ start_node_number
	Generation_increment	integer	[0]	Node generation increment number ≥ 0
	Reference_node_number	integer	[0]	Reference node number ≥ 0
	Low_cut_off_frequency	real	[0.05]	Low cut-off frequency (in Hertz) (unit: 1/T)
	High_cut_off_frequency	real	[1/(2*Dt)]	High cut-off frequency (in Hertz) (unit: 1/T)
	Transition_band_width	real	[□low/2]	Transition band width (in Hertz) (unit: 1/T)

7.5.4 Free-Field Nodes Data

FREE_FIELD_NODES

FREE_FIELD_NODES file_name = "<string>", etc.

This command is used to input free-field nodes data. The free-field motion may be prescribed by either assigning the motion of mesh nodes to be the same as computed in previous free-field computations (see Section 7.5.2), or by using a consistent interface boundary (see Section 9.2.1 and/or Section 9.15.1).

Note	Variable Name	Type	Default	Description
	Type prescribed consistent	list	[none]	Free_field type
	Free_field_motion **displacement** velocity acceleration	list	[displ]	Motion type
	File_name	string	[none]	File name (optional). Name must be enclosed in quotation marks.

• *Prescribed Case:*
Free_field nodes data must then follow in the form (Section 7.5.4.1):
< Node 1, Node 2, ng, (idof(i), i=1, Ndof) >
< etc..., terminate with a blank record >.

• *Consistent Case:*
Free_field nodes data must then follow in the form (Section 7.5.4.2):
< Inside_nodes >
< Node_number, load_time_function_number, (load_dof(i), i=1, Ndof) >
< etc…, terminate with a blank record >
< Outside_nodes >
< Node_number, load_time_function_number, (load_dof(i), i=1, Ndof >
< etc…, terminate with a blank record >.

7.5.4.1 Prescribed Free-Field Nodes Data

Nodes may be assigned to have the same motion as nodes in the free-field (computed previously) for selected degrees of freedom. Such an option is useful for modeling free-field conditions in seismic calculations. Use of this option must be preceded by a READ_MOTION command (see Section 7.5.2).

Note	Variable Name	Type	Default	Description
(1)	Node1	integer	[0]	Node number 1 (node number in the mesh)
	Node2	integer	[0]	Node number 2 (node number in the free-field mesh)
(2)	NG	integer	[0]	Generation increment
	IDOF(1)	integer	[0]	Degree of freedom number
	IDOF(2)	integer	[0]	Degree of freedom number
	.		.	.
	etc.		.	.
	.		.	.
	IDOF(NDOF)	integer	[0]	Degree of freedom number

Notes/
(1) Node number NODE1 is assigned to have the the same motion as node number NODE2 in the free-field, for the degrees of freedom listed on that record. *Terminate with a blank record.*

(2) Free-field condition data can be generated by employing a two record sequence as follows:

Record 1: LODE1, LODE2, LG, IDOF(1),..., IDOF(NDOF)
Record 2: NODE1, NODE2, NG, IDOF(1),..., IDOF(NDOF)

The free-field conditions codes of all nodes:

LODE1 + LG, LODE1 + 2*LG,..., NODE1 - MOD(NODE1 - LODE1, LG)
LODE2 + LG, LODE2 + 2*LG,..., NODE2 - MOD(NODE2 - LODE2, LG)

(i.e., less than NODE1) are set equal to those of node LODE1. If LG is blank or zero, no generation takes place between LODE1 and NODE1.

7.5.4.2 Consistent Free-Field Data

Nodal data are required for both inside and outside nodes. Following the corresponding keyword: "Inside_nodes" or "outside_nodes", nodal data follow as:

Note	Variable Name	Type	Default	Description
	Node_number	integer	[0]	Node number
	ltime	integer	[0]	Load_time function number
	load_dof(i)	real	[0.0]	Multiplier for degree of freedom i (i=1, Ndof)

References / Bibliography

1. Bielak, J. and Christiano, P., "On the Effective Seismic Input for Non-Linear Soil-Structure Interaction Systems," *Earthquake Engineering and Structural Dynamics*, Vol. 12 (1984), 107-119.

2. Cremonini, M, Christiano, P. and Bielak, J., "Implementation of Effective Seismic Input for Soil Structure Interaction Systems," *Earthquake Engineering and Structural Dynamics,* Vol. 16 (1988), 615-625.

8.0 LOAD-TIME FUNCTIONS

There must be at least one load-time function. If no load-time function is specified, the default is a constant = 1.0. The load-time function may be generated by using various procedures as defined hereafter, or may be specified by defining at each time instant its function value. In that latter case, the load-time function is defined by NLS pairs of time instants and function values, where NLS is the number of load steps defined on the control command. A schematic of a typical load-time function is shown in Figure 8.0.1. The time instants must be in ascending order (i.e. $t(j+1) \geq t(j)$, $1 \leq j \leq$ NLS). Load step intervals need not be equal and need not be the same from one load-time function to another. As shown in Figure 8.0.1, the load-time function is assumed to behave in a piecewise linear fashion between data points. For values of t outside the interval [t(1), t(NLS)] the G's are defined by constant extrapolation (i.e. G[t] = G[t(1)] for all $t \leq t(1)$; and G[t] = G[t(NLS)] for all $t \geq t(NLS)$). As an example of the use of this feature, we may take the case in which the load-time function is constant throughout the duration of the analysis. In this case, we may set NLS = 1 and simply read in one data point to define G(t). For instance, the nodal load at time t is defined to be:

$$\mathbf{F}(\mathbf{X}, t) = \sum_{i=1, NLC} \mathbf{F}(\mathbf{X}, i) * G[j, t] \qquad \text{with} \quad j = \text{load\_time\_function}(i), \; i = \text{Load\_case}$$

Element consistent loads (e.g. pressure, gravity, etc.) are also multiplied by load-time functions. The load case number is defined in the element group data (see Chapter 9). Input for a load-time function is described below.

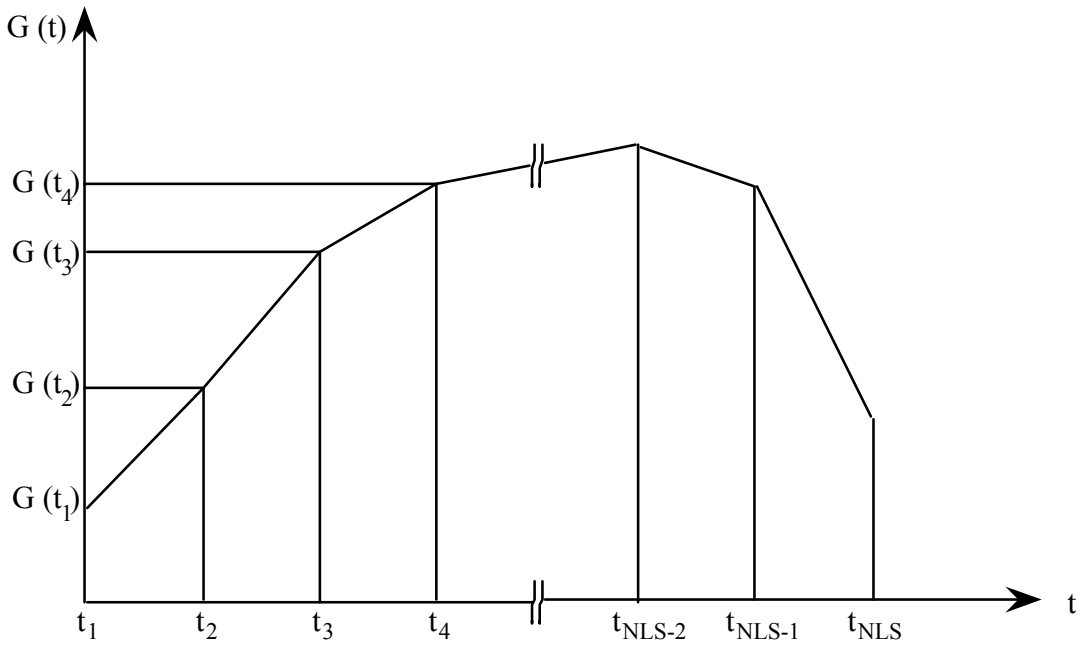


Figure 8.0.1 Schematic Representation of a Load-Time Function

8.1 Load Time Functions

LOAD_TIME_FUNCTION

LOAD_TIME_FUNCTION function_type = type , etc...
< etc..., terminate with a blank record >

Create a load-time function. Two main options are available. The load-time function may be read in directly as a list (optionally from another file) or may be generated.

Note	Variable Name	Type	Default	Description
	Load_time_function_number	integer	[1]	Load time function number
	Function_type **piecewise_linear** piecewise_constant function_formula acceleration_spectrum design_spectrum	list	[*]	Function type
• *Piecewise Linear and Piecewise Constant Cases*				
	Time_offset	real	[0.0]	Time offset
(1)	File_name	string	[none]	File name (optional). Name must be enclosed in quotation marks.
	Load time function data must then follow in the form: < Time_instant (i), Load_time_function_value (i) > (i = 1, number_of_load_steps) < etc..., terminate with a blank record >			
• *Function Formula Case*				
(2)	A_1 , A_2 , ... , A_7	real	[0.0]	Formula coefficients
• *Acceleration Spectrum Case*				
(3)	Duration	real	[10.0]	Duration (in time units: T)
	Cut_off_freq	real	[10.0]	Cut-off frequency (in Hertz: 1/T)
	Max_acceleration	real	[0.20]	Maximum acceleration (in units: L/T^2)
	Damping_ratio	real	[0.05]	Damping ratio; > 0.0 and < 1.0
	Seed	real	[7654321]	Seed for random number generation; > 100,000 and < 100,000,000
	Spectral_ordinate_1	real	[0.5]	Spectral ordinate S_1 at period T_1
	Spectral_abscissa_1	real	[0.15625]	Spectral abscissa T_1
	Spectral_ordinate_2	real	[0.5]	Spectral ordinate S_2 at period T_2
	Spectral_abscissa_2	real	[0.40]	Spectral abscissa T_2
	Spectral_ordinate_3	real	[0.2]	Spectral ordinate S_3 at period T_3
	Spectral_abscissa_3	real	[1.0]	Spectral abscissa T_3
	Power_exponent	real	[1.0]	Power exponent m
(4)	Rise_time_t_1	real	[2.0]	Rise time t_1
	Decay_time_t_2	real	[5.0]	Decay time t_2
	Decay_rate_c	real	[0.4]	Decay rate c

(cont'd)

(cont'd)

Note	Variable Name	Type	Default	Description
• *Design Spectrum Case*				
(5)	Damping_ratio	real	[0.05]	Damping ratio; > 0.0 and < 1.0
	Max_acceleration	real	[0.20]	Maximum acceleration (in units: L/T^2)
	Spectral_ordinate_1	real	[0.5]	Spectral ordinate S_1 at period T_1
	Spectral_abscissa_1	real	[0.15625]	Spectral abscissa T_1
	Spectral_ordinate_2	real	[0.5]	Spectral ordinate S_2 at period T_2
	Spectral_abscissa_2	real	[0.40]	Spectral abscissa T_2
	Spectral_ordinate_3	real	[0.2]	Spectral ordinate S_3 at period T_3
	Spectral_abscissa_3	real	[1.0]	Spectral abscissa T_3
	Power_exponent	real	[1.0]	Power exponent m

EXAMPLE

```
Load_Time_Function   /
  load_time_function_number = 1 /
  function_type = piecewise_linear
    0.0 , 1.0                          # at time = 0.0, function = 1.0
    1.0 , 5.0                          # at time = 1.0, function = 5.0
```

Notes/

(1) This option allows the load-time function to be read in from a data file separate from the main input file. The default (i.e., filename is left empty) assumes that the load-time function is contained in the main input file.

(2) The load time-function is generated by using the following formula:

$$G(t) = A_1 + A_2 t$$

If ($A_3 A_4 \neq 0.0$) then:

$$G(t) = A_1 + A_2 t + A_3 \left[\sin\left(A_4 (t - A_5)\right)\right]^{A_6} \text{ when } A_5 \leq t \leq A_7$$

and when $t \geq A_7$ then:

$$G(t) = G(A_7) = A_1 + A_2 A_7 + A_3 \left[\sin\left(A_4 (A_7 - A_5)\right)\right]^{A_6} \text{ when } t \geq A_7$$

where the parameters A_1,..., A_7 are input.

(3) The load time-function is assumed to be an acceleration time history. The acceleration time history is generated by using the acceleration response spectrum shown in Figure 8.1.1, following the procedure proposed by Vanmarcke et al. (1976) and reported in [1]. The implementation of the procedure was performed in cooperation with Dr. G. Deodatis at Princeton University in the Fall of 1992. Note that for the UBC (1994) spectrum, $S_1 = 2.5\ A_{max}$, $S_2 = S_1$ and $S_3 T_3 = S_2 T_2$ (i.e., m=1).

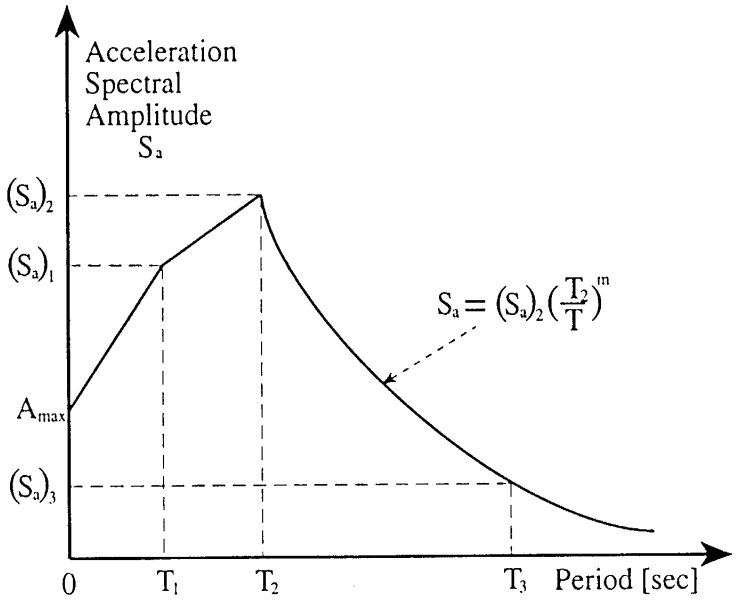


Figure 8.1.1 Acceleration Spectrum

(4) The acceleration time history envelope is given by the following formula (Jennings, 1968):

$$\text{acc} = A_{max} \quad t^2 / t_1^2 \qquad 0 \le t \le t_1$$
$$\text{acc} = A_{max} \qquad t_1 \le t \le t_2$$
$$\text{acc} = A_{max} \quad \exp\left(-c\,(t - t_2)\right) \qquad t \ge t_2$$

as illustrated in Figure 8.1.2

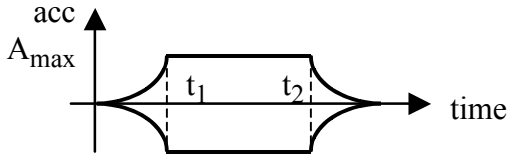


Figure 8.1.2

(5) The load time-function is an acceleration spectrum as shown in Figure 8.1.1. Used for spectral analysis (see Section 12.2).

Reference / Bibliography

1. Gasparini, D.A. and E.H. Vanmarcke,"Simulated Earthquake Motions Compatible with Prescribed Response Spectra", *MIT Report No. R76-4*, (Jan. 1976).

8.2 Filtering Request

FILTER

FILTER load_time_function_number = ltime, etc...

This allows input load-time functions to be filtered.

Note	Variable Name	Type	Default	Description
	Load_time_function_number	integer	[0]	Load-time function number
	Low_cut_off_frequency	real	[0.05]	Low cut-off frequency (in Hertz) (unit: 1/T)
	High_cut_off_frequency	real	[1/(2*Dt)]	High cut-off frequency (in Hertz) (unit: 1/T)
	Transition_band_width	real	[□low/2]	Transition band width (in Hertz) (unit: 1/T)

Notes . .

9.0 ELEMENT DATA

9.1 Element Control Information

9.1.1 Define a Group of Elements

DEFINE_ELEMENT_GROUP

DEFINE_ELEMENT_GROUP name = " <string>" etc...

The elements may be read in groups. An element group is a collection of finite elements which model the same governing balance equation(s) within a region. This command must precede the element group data, and defines the element group control parameters as follows.

Note	Variable Name	Type	Default	Description
	Name	string	[none]	Element group name. Name must be enclosed in quotation marks.
(1)	Group_number	integer	[*]	Element group number (optional) (Only active if Ndeath > Nbirth)
(2)	Multi_processing on / **off**	list	[off]	Multi-processing option
	Element_type continuum interface structural nodal	list	[none]	Element type
	Element_shape eight_node_brick twenty_node_brick six_node_wedge fifteen_node_wedge four_node_tetra ten_node_tetra four_node_quad eight_node_quad nine_node_quad three_node_tri six_node_tri two_node_line three_node_line one_node	list	[none]	Element shape

(cont'd)

(cont'd)

Note	Variable Name	Type	Default	Description
	Element_name	list	[none]	Element name:
• *Continuum Type Elements*				
	QDC_Solid			See Section 9.2.0.1
	QDC_Fluid			See Section 9.2.0.2
	QDC_Stokes			See Section 9.2.0.3
	QDC_Transport			See Section 9.2.0.4
	QDC_Helmoltz			See Section 9.2.0.5
	QDC_Ale			See Section 9.2.0.6
	QDC_Thermal			See Section 9.2.0.7
	QDC_Heat			See Section 9.2.0.8
	QDC_Transport			See Section 9.2.0.9
	QDC_Charge			See Section 9.2.0.10
	QDC_Porous			See Section 9.2.0.11
	QDC_Darcy			See Section 9.2.0.12
	QDC_Pressure			See Section 9.2.0.13
	cmi_QDCA			See Section 9.2.0.14
	QDC_MFlow			See Section 9.2.0.15
	QDCP_Mixed			See Section 9.2.0.16
	cmi_QDCS			See Section 9.2.0.17
	cmi_QDCP_Pressure			See Section 9.2.0.18
	cmi_QDCZ			See Section 9.2.0.19
	Multi_Phase_Heat			See Section 9.2.0.20
	QDC_Laplace			See Section 9.2.0.21
	QDC_Stream			See Section 9.2.0.22
	QDC_Level			See Section 9.2.0.23
• *Structural Type Elements*				
	Linear_Truss			See Section 9.3.0.1.1
	Nonlinear_Truss			See Section 9.3.0.1.2
	Linear_Beam			See Section 9.3.0.2.1
	Nonlinear_Beam			See Section 9.3.0.2.2
	Plate			See Section 9.3.0.3.1
	Shell_Plate			See Section 9.3.0.3.2
	Shell_Bilinear			See Section 9.3.0.3.3
	Membrane			See Section 9.3.0.4
• *Interface Type Elements*				
	Interface_Surface			See Section 9.4
	Contact_Surface			See Section 9.5
	Slide_Line			See Section 9.6
	Slide_Coulomb			See Section 9.7
	Contact_Plane			See Section 9.8
	Crack_Xfem			See Section 9.9
	Level_Xfem			See Section 9.10
	Multi_Point_Constraint			See Section 9.11

(cont'd)

(cont'd)

Note	Variable Name	Type	Default	Description
• *Nodal Type Elements*				
	Nodal_Mass			See Section 9.11.1
	Nodal_Damping			See Section 9.11.2
	Nodal_Spring			See Section 9.11.3
	Nodal_Reaction			See Section 9.12
	Nodal_Link			See Section 9.13
	Nodal_Penalty			See Section 9.14
	Nodal_Transmitting			See Section 9.15
	Number_of_output_sets	integer	[0]	Number of field output histories
	Spatial_output **on** / off	list	[on]	Spatial field output
(3)	Material_model	string	[none]	Name of material model to apply to this element group. Name must be enclosed in quotation marks.
(4)	Number_of_material_sets	integer	[1]	Number of material sets ≥ 1 (Only required if material_model is not specified.)
	Number_of_geometric_sets	integer	[0]	Number of geometric sets
(5)	Number_of_phases	integer	[1]	Number of fluid phases
(5)	Number_of_components	integer	[1]	Number of components

EXAMPLE

```
Define_Element_Group  /
    name = "group_name"  /
    group_number = 1  /
    element_type = continuum  /
    element_shape = four_node_quad  /
    material_model = "dummy" /
    element_name = QDC_Porous  /
        analysis_type = axisymmetric  /
        strain_displacement = bbar_mean  /
        number_of_output_sets = 1
```

Notes/

(1) The element group number ("Neg") may be specified. If not specified, then Neg=Neg+1, where Neg is the number of the last element group entered.

(2) This option allows multiprocessing of elemental arrays (e.g., using PVM and/or vectorizations options).

(3) The material model's name must have been previously defined by the command DEFINE_MATERIAL_MODEL (see Section 10.0.1). Option restricted to continuum and structural type elements.

(4) This option allows the material data to be specified within the element group data.

(5) Only applicable to multi-phase problems.

9.1.2 Define a Region

DEFINE_REGION

DEFINE_REGION name = "<string>" element_type = ... , etc...

A region is a collection of elements within the finite element mesh. Each region describes a spatial domain within the problem. Within a region, several element groups may be specified to model different governing balance equation(s) (e.g., by specifying that both the solid equation QDC_Solid, and the scalar diffusion equation QDC_Pressure be active within a region to model a saturated porous medium). If within a region more than one equation type is specified, then each corresponding element group must be assigned to a different solution stagger (see Section 2.1). After defining a region, the required governing equations within the region must be defined by specifying one or more element groups (see Section 9.1.2.1).

Note	Variable Name	Type	Default	Description
	Name	string	[none]	Region name. Name must be enclosed in quotation marks.
	Element_type continuum structural	list	[none]	Element type
	Element_shape eight_node_brick twenty_node_brick six_node_wedge fifteen_node_wedge four_node_tetra ten_node_tetra four_node_quad eight_node_quad nine_node_quad three_node_tri six_node_tri two_node_line three_node_line one_node	list	[none]	Element shape
	Analysis_type one_dimensional **plane** axisymmetric three_dimensional	list	[plane]	Analysis type: One-dimensional Two-dimensional / plane strain Axisymmetric Three-dimensional

(cont'd)

(cont'd)

Note	Variable Name	Type	Default	Description
(1)	Multi_processing on / **off**	list	[off]	Multi-processing option
	Number_of_output_sets	integer	[0]	Number of field output histories
	Spatial_output **on** / off	list	[on]	Spatial field output
(2)	Material_model	string	[none]	Name of material model to apply to this region. Name must be enclosed in quotation marks.
(3)	Number_of_material_sets	integer	[1]	Number of material sets ≥ 1 (Only required if material_model is not specified.)
(4)	Number_of_phases	integer	[1]	Number of fluid phases
(4)	Number_of_components	integer	[1]	Number of components

EXAMPLE

```
Define_Region  /
    name = "region_name"  /
    element_type = continuum  /
    element_shape = four_node_quad  /
    analysis_type = axisymmetric  /
    number_of_output_sets = 1  /
    material_model = "dummy"

    Element_Group  /
        name = "stress_equation"  /
        element_name = QDC_solid  /
        strain_displacement = bbar_mean

    Element_Group  /
        name = "pressure_equation"  /
        element_name = QDC_pressure
```

Notes/
(1) This option allows multiprocessing of elemental arrays (e.g., using PVM and/or vectorizations options).

(2) The material model's name must have been previously defined by the command DEFINE_MATERIAL_MODEL (see Section 10.0.1). Option restricted to continuum and structural type elements.

(3) This option allows the material data to be specified within the region.

(4) Only applicable to multi-phase problems.

9.1.2.1 Element Group

ELEMENT_GROUP

ELEMENT_GROUP name = "<string>" group_number = neg , etc...

The command is used to specify a governing balance equation(s) within a region.

Note	Variable Name	Type	Default	Description
	Name	string	[none]	Element group name. Name must be enclosed in quotation marks.
(1)	Group_number	integer	[*]	Element group number (optional)
	Element_name	list	[none]	Element name:

• *Continuum Type Elements*

QDC_Solid	See Section 9.2.0.1
QDC_Fluid	See Section 9.2.0.2
QDC_Stokes	See Section 9.2.0.3
QDC_Transport	See Section 9.2.0.4
QDC_Helmoltz	See Section 9.2.0.5
QDC_Ale	See Section 9.2.0.6
QDC_Thermal	See Section 9.2.0.7
QDC_Heat	See Section 9.2.0.8
QDC_Transport	See Section 9.2.0.9
QDC_Charge	See Section 9.2.0.10
QDC_Porous	See Section 9.2.0.11
QDC_Darcy	See Section 9.2.0.12
QDC_Pressure	See Section 9.2.0.13
cmi_QDCA	See Section 9.2.0.14
QDC_MFlow	See Section 9.2.0.15
QDCP_Mixed	See Section 9.2.0.16
cmi_QDCS	See Section 9.2.0.17
cmi_QDCP_Pressure	See Section 9.2.0.18
cmi_QDCZ	See Section 9.2.0.19
Multi_Phase_Heat	See Section 9.2.0.20
QDC_Laplace	See Section 9.2.0.21
QDC_Stream	See Section 9.2.0.22
QDC_Level	See Section 9.2.0.23

(cont'd)

(cont'd)

Note	Variable Name	Type	Default	Description
• *Structural Type Elements*				
	Linear_Truss			See Section 9.3.0.1.1
	Nonlinear_Truss			See Section 9.3.0.1.2
	Linear_Beam			See Section 9.3.0.2.1
	Nonlinear_Beam			See Section 9.3.0.2.2
	Plate			See Section 9.3.0.3.1
	Shell_Plate			See Section 9.3.0.3.2
	Shell_Bilinear			See Section 9.3.0.3.3
	Membrane			See Section 9.3.0.4

Notes/

(1) The element group number ("Neg") may be specified. If not specified, then Neg=Neg+1, where Neg is the number of the last element group entered.

9.2 Continuum Elements

9.2.0 Analysis Options

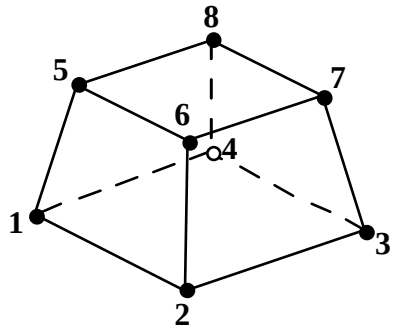
In one dimension the element is defined by 2 nodes. If the number of spatial dimension NSD = 1 (defined in Section 2.1), usual rod theory is used, and the element is assumed to have axial kinematics only. Otherwise, i.e. if NSD > 1, the element is assumed to be aligned with the x_2-axis in the global reference frame (x_1, x_2, x_3), and NSD kinematics in directions x_1,x_2, etc., are assigned to each node.

In two dimensions the element may be used in triangular (3 node or 6 node) or quadrilateral (4 node, 8 node or 9 node) form for plane and axisymmetric analysis. The nodes of the element must be input in counterclockwise order in the order shown in Figure 9.2.0.1. The plane of analysis is assumed to be the x_1,x_2 plane, and the element is assumed to have unit thickness in the plane option. In axisymmetric analysis the radial direction is specified as the x_1-axis. Reduced / selective numerical integration and the mean dilatational formulation may be employed to improve element behavior in various situations. *These options should be activated only by users fully knowledgeable in their use.*

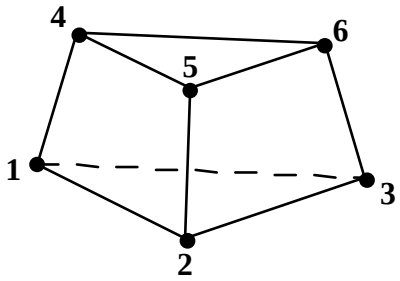
In three dimensions the element may be used in tetrahedral (10 or 4 node), wedge (6 node or 15 node) or brick (8 node or 20 node) form. The nodes of the element must be input in the order shown in Figure 9.2.0.1. Reduced/selective numerical integration and the mean dilatational formulation may be employed to improve element behavior in various situations. *These options should be activated only by users fully knowledgeable in their use.*

Stresses/strains in the global coordinate system, principal stresses/strains, maximum shear stress/strain and angle of inclination, in degrees, of principal states are output at the element centroid, which is generally the point of optimal accuracy. All shear strains are reported according to the "engineering" convention (i.e. twice the value of the tensor components).

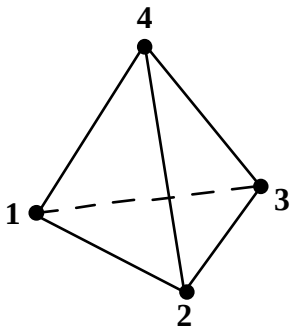
In the following, NSD = number of spatial dimension; NDOF = total number of degrees of freedom; and NED = element nodal local degrees of freedom.



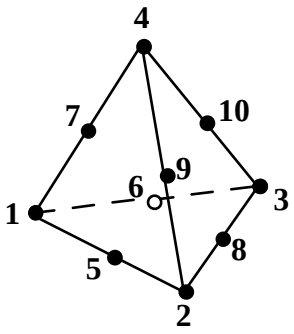
8-node brick



6-node wedge

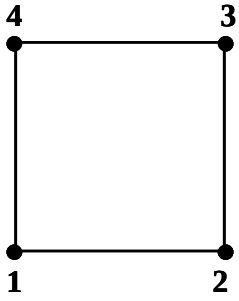


4-node tetrahedron

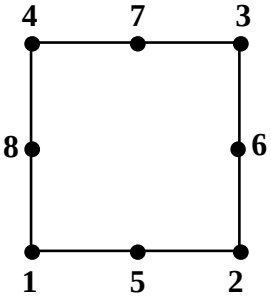


10-node tetrahedron

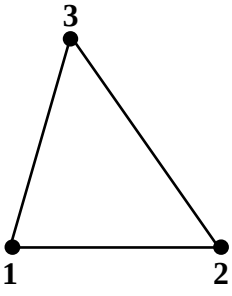
3D CONTINUUM



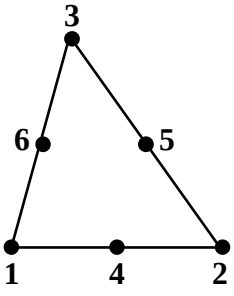
4-node quadrilateral



8-node quadrilateral

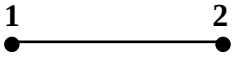


3-node triangle

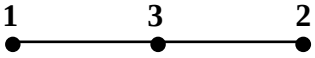


6-node triangle

2D CONTINUUM



2-node line



3-node line

1D CONTINUUM

Figure 9.2.0.1 Continuum Elements

9.2.0.1 Solid Equation

QDC_SOLID

Element_name = QDC_Solid, etc…
< stress material data >
< body force data >
< connectivity data >
< field output data >

The element is used for solution of the following equations:

$$\nabla \cdot \underset{\sim}{\sigma} + \rho \underset{\sim}{b} = \rho \underset{\sim}{a}$$

where $\underset{\sim}{\sigma}$ = solid stress, $\underset{\sim}{a}$ = solid acceleration, $\underset{\sim}{b}$ = body force (per unit mass) and ρ = mass density. NSD solid kinematic degrees of freedom are assigned to each node, in the $x_1, x_2, (\text{and} x_3)$ directions, respectively.

- For saturated porous media applications

  $$\underset{\sim}{\sigma} = \underset{\sim}{\sigma}'^{s} - b\, p^{w} \underset{\sim}{\delta}$$

  where $\underset{\sim}{\sigma}'^{s}$ = solid effective stress, p^{w} = pore fluid pressure; $b = 1 - C_s / C_m$; C_s = solid grains compressibility; C_m = solid matrix compressibility; $\rho = \rho^{s} + \rho^{w}$ = total mass density, $\rho^{s} = (1-\varphi)\,\rho_s$ and $\rho^{w} = \varphi\,\rho_w$ with ρ_s = solid mass density, ρ_w = fluid mass density and φ = porosity. For fully undrained (viz., no diffusion) cases, the pore fluid pressure is determined from the computed solid velocities through the following equation:

  $$p^{w} = -b\, M\, \nabla \cdot \underset{\sim}{v}$$

  where $\frac{1}{M} = \frac{b - \phi_0}{K_s} + \frac{\phi}{K_f}$; K_s = solid grains modulus, K_f = fluid bulk modulus

- For multi-phase fluid flow applications:

  $$\underset{\sim}{\sigma} = \underset{\sim}{\sigma}'^{s} - b \left(\sum_{\beta} S_\beta p_\beta \right) \underset{\sim}{\delta}$$

  where $\underset{\sim}{\sigma}'^{s}$ = solid effective stress; β = fluid phase number (β = 1, number_of_phases); S_β = degree of saturation; p_β = fluid pressure;

$\rho = \rho^s + \rho^f$ = total mass density; $\rho^s = (1-\varphi)\rho_s$ with ρ_s = solid mass density and φ = porosity; $\rho^f = \varphi \sum_\beta \rho_\beta S_\beta$ with ρ_β = fluid phase mass density.

- For thermo solids:

$$\underset{\sim}{\sigma} = \underset{\sim}{\sigma}'^s - \underset{\sim}{\beta}\left(\theta - \theta_o\right)$$

where $\underset{\sim}{\sigma}'^s$ = solid effective stress, θ = temperature and $\underset{\sim}{\beta}$ = thermal moduli (second-order tensor).

- For piezoelectric solids:

$$\underset{\sim}{\sigma} = \underset{\sim}{\sigma}'^s - \underset{\sim}{e} \cdot \underset{\sim}{E}$$

where $\underset{\sim}{\sigma}'^s$ = solid effective stress, $\underset{\sim}{E}$ = electric field (viz., $\underset{\sim}{E} = -\nabla\varphi$ where φ = electric potential), and $\underset{\sim}{e}$ = piezoelectric constants (third-order tensor, viz., $\left(\underset{\sim}{e} \cdot \underset{\sim}{E}\right)_{ij} = e_{kij}E_k$).

Implementation Issues

For coupled problems, in the implementation we adopted, the dependent variables are the velocity and the pressure and/or temperature, and a Petrov-Galerkin formulation is used to circumvent restrictions of the Babuska-Brezzi condition. In particular, equal-order interpolations are used for both the velocity and the pressure.

References / Bibliography

1. Babuska, I., "Error Bounds for Finite Element Method," *Numer. Math,* 16 (1971), 322-333.

2. Brezzi, F..,"On the Existence, Uniqueness and Approximation of Saddle-Point Problems Arising from Lagrange Multipliers," *Rev. Francaise d'Automatique Inform. Rech. Oper.,* Ser. Rouge Anal. Numer. 8, R-2 (1974) 129-151.

3. Hughes, T.J.R, Franca, L.P. and Balestra, M., "A New Finite Element Formulation for Computational Fluid Dynamics: V. Circumventing the Babuska-Brezzi Condition: A Stable Petrov-Galerkin Formulation of the Stokes Problem Accommodating Equal-Order Interpolations," *Comp. Meth. Appl. Mech. Eng.,* Vol. 59 (1986) 85-99.

9.2.0.2 Fluid Equation

QDC_FLUID

Element_name = QDC_Fluid , etc…
< stress material data >
< body force data >
< connectivity data >
< field output data >

The element is used for solution of the following equations:

$$\rho\left(\frac{\partial \underset{\sim}{v}}{\partial t}\right)+\rho\underset{\sim}{v}\cdot\nabla\underset{\sim}{v}=\nabla\cdot\underset{\sim}{\sigma}+\rho\underset{\sim}{b}$$

where $\underset{\sim}{\sigma}=$ stress, $\underset{\sim}{v}=$ velocity, $\underset{\sim}{b}=$ body force (per unit mass), $\rho=$ mass density. The fluid stress is given by the following equation:

$$\underset{\sim}{\sigma}=-p\underset{\sim}{\delta}+\underset{\sim}{\tau}$$

where p is the pressure, and $\underset{\sim}{\tau}$ the viscous stress tensor. For isotropic fluids the following viscous relation is used:

$$\underset{\sim}{\tau}=\lambda^{w}\nabla\cdot\underset{\sim}{v}\underset{\sim}{\delta}+2\mu^{w}\underset{\sim}{v}_{(\,)}$$

where

$\underset{\sim}{v}_{(\,)}=$ symmetric part of the velocity gradient, i.e., in components form:

$$v_{(i,j)}=\frac{1}{2}\left(v_{i,j}+v_{j,i}\right)=\dot{\varepsilon}_{ij}$$

and

$$\nabla\cdot\underset{\sim}{v}=tr\left(v_{i,j}\right)=v_{i,i}=\dot{\varepsilon}_{ii}$$

In the above equation λ^{w} and μ^{w} are the viscosity coefficients (also referred to as the Lamé coefficients). For incompressible flows, $\nabla\cdot\underset{\sim}{v}=\dot{\varepsilon}_{ii}=0$. In our implementation, isothermal incompressible and "slightly compressible" flows are considered.

For *incompressible* flows, the pressure is determined from the computed velocities through the following equation:

$$p=-\lambda\nabla\cdot\underset{\sim}{v}$$

where $\lambda=$ a penalty parameter. Clearly, λ must be large enough so that the compressibility and pressure errors are negligible, yet not so large that numerical ill-conditioning ensues. The criterion used is (see [1]):

$$\lambda = \mathrm{c}\, max\left(\mu^{\mathrm{w}}, \mu^{\mathrm{w}} \mathrm{R}_{\mathrm{e}}\right)$$

where c is a constant which depends only on the computer word length (it is independent of the mesh parameter h) and R_e is the Reynolds number. Numerical studies reveal that for floating-point word lengths of 60 to 64 bits, $c \approx 10^7$. The Reynolds number is computed as

$$R_e = \rho \frac{VL}{\mu^w}$$

where V and L are "characteristic" velocity and length, respectively. The characteristic velocity V is usually taken to be the maximum expected velocity in the flow, and L is taken as a major dimension of the problem (e.g., the diameter).

For "*slightly compressible"* flows, the pressure is determined through the following equation:

$$\frac{\partial \mathrm{p}}{\partial \mathrm{t}} = -\lambda \nabla \cdot \underset{\sim}{v}$$

where $\lambda = \rho \dfrac{\partial \mathrm{p}}{\partial \rho} =$ fluid bulk modulus.

Implementation Issues

In the implementation we adopted, the convective force is treated explicitly (to avoid non-symmetric matrices). This engenders some stability restrictions. Stability analyses indicate that if no iterations are performed, the upwind scheme used (see Section 9.2.1) for the convection are stable if Δt satisfies a Courant condition [1], viz.,

$$\Delta t \leq \frac{h}{\left\| \underset{\sim}{v} \right\|}$$

where h = mesh size parameter, and $\left\| \underset{\sim}{v} \right\| =$ velocity magnitude for the element. The above inequality must be satisfied for each finite element in the mesh. (Note it is solely a convection condition and in particular is independent of the Reynolds number.)

For fluid applications, NSD fluid kinematic degrees of freedom are assigned to each node, in the $x_1, x_2, (\text{and} x_3)$ directions, respectively. The time integration must be performed with a hyperbolic time integrator (see Section 12.2) as the resulting initial boundary value problem is treated as *parabolic-hyperbolic*.

Remark: For isothermal incompressible flows, there is no energy balance or temperature equation. Thus the pressure variable enters the system of equations only through its derivative and therefore p can be determined only up to an arbitrary constant. This means that the pressure must be specified externally somewhere in the flow field.

References / Bibliography

1. Hughes et al., "Review of Finite Element Analysis of Incompressible Viscous Flows by the Penalty Function Formulation," *J. Comp. Phys.*, 30, No. 1, (1979), 1-60.

2. Hughes, T.J.R., et al.,"Lagrangian-Eulerian Finite Element Formulation for Incompressible Flows," *Comp. Meth. Appl. Mech. Eng.*, Vol. 29, (1981), 329-349.

3. Prevost, J.H., and Hughes, T.J.R.,"Dynamic Fluid-Structure-Soil Interaction," *ASCE Publication on Geotechnical Practice in Offshore Engineering*, (1983), 133-143.

9.2.0.3 Stokes Flow Equation

QDC_STOKES

Element_name = QDC_Stokes , etc…
< stress material data >
< body force data >
< connectivity data >
< field output data >

The element is used for solution of the following equations:

$$\nabla \cdot \underset{\sim}{\sigma} + \rho \underset{\sim}{b} = 0 \qquad \text{(momentum balance)}$$
$$\nabla \cdot \underset{\sim}{u} = 0 \qquad \text{(incompressibility condition)}$$

where $\underset{\sim}{\sigma}$ = Cauchy stress, $\underset{\sim}{b}$ = body force (per unit mass), ρ = mass density. The fluid stress is given by the following equation:

$$\underset{\sim}{\sigma} = -p\underset{\sim}{\delta} + 2\mu\, \underset{\sim}{\varepsilon}(\underset{\sim}{u})$$

where p is the pressure, and μ is the dynamic viscosity; $\underset{\sim}{u}$ is the velocity vector, and $\underset{\sim}{\varepsilon}(\underset{\sim}{u})$ is the symmetrical part of the velocity gradient.

Implementation Issues

In the implementation we adopted, the dependent variables are the velocity and the pressure, and a Petrov-Galerkin formulation is used to circumvent restrictions of the Babuska-Brezzi condition. In particular, equal-order interpolations are used for both the velocity and the pressure.

References / Bibliography

1. Babuska, I., "Error Bounds for Finite Element Method," *Numer. Math*, 16 (1971), 322-333.

2. Brezzi, F..,"On the Existence, Uniqueness and Approximation of Saddle-Point Problems Arising from Lagrange Multipliers," *Rev. Francaise d'Automatique Inform. Rech. Oper.*, Ser. Rouge Anal. Numer. 8, R-2 (1974) 129-151.

3. Hughes, T.J.R, Franca, L.P. and Balestra, M., "A New Finite Element Formulation for Computational Fluid Dynamics: V. Circumventing the Babuska-Brezzi Condition: A Stable Petrov-Galerkin Formulation of the Stokes Problem Accommodating Equal-Order Interpolations," *Comp. Meth. Appl. Mech. Eng.*, Vol. 59 (1986) 85-99.

9.2.0.4 Scalar Convection-Diffusion Equation

QDC_TRANSPORT

Element_name = QDC_Transport , etc…
< scalar diffusion material data >
< body force data>
< connectivity data >

The element is used for solution of the following scalar convection-diffusion equation:

$$\rho \frac{\partial \varphi}{\partial t} + \rho \, \underset{\sim}{v} \cdot \nabla \varphi - \nabla \cdot \left[\underset{\sim}{k} \cdot \nabla \varphi \right] = b$$

where $\varphi =$ concentration, $\rho =$ mass density, $\underset{\sim}{k} =$ diffusivity = diffusion/dispersion coefficient matrix, $b =$ body force, and $\underset{\sim}{v} =$ given flow velocity. The Peclet number is defined as follows:

$$P_e = \rho \left\| \underset{\sim}{v} \right\| L / k$$

where $L =$ characteristic length. For $P_e = 0$ a purely diffusive solution is obtained, whereas for $P_e = \infty$ solution to the first-order wave equation is obtained. For convection-diffusion (parabolic mode) and advection-diffusion (elliptic mode) one degree of freedom is assigned to each node for the concentration ϕ. In the implementation we adopted, the connective force is treated *implicitly* and a *non-symmetric* linear-solver must therefore be used (see Section 12.4). A stabilized SUPG formulation is used.

References / Bibliography

1. Brooks, A.N. and Hughes, T.J.R., "Streamline Upwind/Petrov-Galerkin Formulations for Convection Dominated Flows with Particular Emphasis on the Incompressible Navier-Stokes Equations," *Computer Methods in Applied Mechanics and Engineering*, Vol. 32, (1982), pp. 199-259.

9.2.0.5 Helmoltz/Laplace Equation

QDC_HELMOLTZ

Element_name = QDC_Helmoltz , etc…
< material data >
< connectivity data >

The element is used for solution of the following scalar equation:

$$\nabla^2 p = \frac{1}{c^2}\frac{\partial^2 p}{\partial t^2}$$

where p = pressure and c = wave speed $\left(= \sqrt{\frac{\lambda + 2\mu}{\rho}},\ \lambda, \mu = \text{Lame's elastic moduli} \right)$.

One degree of freedom is assigned to each node for the pressure. The element may be used to solve the Laplace equation (elliptic mode) and the Helmoltz equation (hyperbolic mode).

9.2.0.6 Mesh Motion Equation

QDC_ALE

Element_name = QDC_ale , etc…
< connectivity data >

The element is used to compute the mesh displacement field in arbitrary Lagrangian-Eulerian (ALE) models (see Section 5.4), by solving the following vector Laplace equation:

$$\nabla \cdot \left[(1+\tau) \nabla \underset{\sim}{U} \right] = \underset{\sim}{0}$$

subject to appropriate prescribed displacement boundary conditions at the moving boundary. The moving boundary is composed of the moving fluid-solid interfaces as well as the oscillating free surfaces. The parameter τ is a bounded, nondimensional function designed to prevent the inversion of small elements (see e.g. [1]) as follows. For each element τ is defined as:

$$\tau^e = \frac{\Delta_{\max} - \Delta_{\min}}{\Delta_e} \qquad e = 1, 2, \ldots, \text{numel} \tag{1}$$

where $\Delta^e =$ area (or volume in 3D) of the current element,

$$\max_e \left\{ \tau^e \right\} = \frac{\Delta_{\max} - \Delta_{\min}}{\Delta_{\min}} = M < \infty \tag{2}$$

$$\min_e \left\{ \tau^e \right\} = \frac{\Delta_{\max} - \Delta_{\min}}{\Delta_{\max}} = m > 0 \tag{3}$$

Remarks: For the degenerate case of a uniform mesh, $\Delta_{\max} = \Delta_{\min} = \Delta^e$ and the Laplace equation works well without the additional constraint equation over the element, and $\tau = 0$.

References / Bibliography

1. Masud, A. and T.J.R. Hughes, "A Space-time Finite Element Method for Fluid-structure Interaction," *SUDAM Report No. 93-3*, Stanford University, Stanford, CA, (1993).

Notes . .

9.2.0.7 Coupled Thermo-Solid Equation

QDC_THERMAL

Element_name = QDC_Thermal , etc…
< stress material data >
< heat conduction material data >
< body force data>
< connectivity data >
< field output data >

For thermal applications, NED = (NSD+1) degrees of freedom are assigned to each node. The first NSD degrees of freedoms are solid kinematic degrees of freedom, in the $x_1, x_2, (\text{and} x_3)$ directions, and the last degree of freedom is assigned to the temperature.

In the *parabolic mode* (diffusion analysis), the element is used for solution of the following coupled equations:

$$-\nabla \cdot \left[\underset{\sim}{\sigma} - \underset{\sim}{\beta}\theta\underset{\sim}{\delta} \right] = \rho \underset{\sim}{b}$$

$$\rho \, \mathrm{c} \, \frac{d\theta}{dt} + \mathrm{T}_0 \underset{\sim}{\beta} \cdot \nabla \underset{\sim}{v} - \nabla \cdot \left[\underset{\sim}{k} \cdot \nabla \theta \right] = \rho \, h$$

where $\underset{\sim}{\sigma} =$ solid stress, $\underset{\sim}{v} =$ solid velocity, $\underset{\sim}{b} =$ body force $\underset{\sim}{\beta} =$ thermal moduli, $\theta =$ temperature, T_0 = reference temperature, $\rho =$ mass density, $c =$ specific heat, $\underset{\sim}{k} =$ thermal conductivity, $h =$ heat source.

In the *hyperbolic mode* (dynamic analysis), the element is used for solution of the following coupled equations:

$$\rho \underset{\sim}{a} - \nabla \cdot \left[\sigma - \underset{\sim}{\beta}\theta\underset{\sim}{\delta} \right] = \rho \underset{\sim}{b}$$

$$\rho \, c \, \frac{d\theta}{dt} + \mathrm{T}_0 \underset{\sim}{\beta} \cdot \nabla \underset{\sim}{v} - \nabla \cdot \left[\underset{\sim}{k} \cdot \nabla \theta \right] = \rho \, h$$

where $\underset{\sim}{\sigma} =$ solid stress, $\underset{\sim}{a} =$ solid acceleration, $\underset{\sim}{v} =$ solid velocity, $\underset{\sim}{b} =$ body force, $\underset{\sim}{\beta} =$ thermal moduli, $\theta =$ temperature, T_0 = reference temperature, $\rho =$ mass density, $c =$ specific heat, $\underset{\sim}{k} =$ thermal conductivity, $h =$ heat source.

References / Bibliography

1. Prevost, J.H., and Tao, D.J., "Finite Element Analysis of Dynamic Coupled Thermoelastic Problems with Relaxation Times," *J. Appl. Mech.*, ASME, Vol. 50, (1983), pp. 817-822.

2. Tao, D.J., and Prevost, J.H., "Relaxation Effects on Generalized Thermoelastic Waves," *J. Thermal Stresses,* Vol. 7, (1984), pp. 79-89.

3. Tao, D.J., "Finite Element Analysis of Thermoelasticity Problems," Ph.D. Thesis, Department of Civil Engineering, Princeton University, Princeton, New Jersey, (June 1984).

9.2.0.8 Heat Equation

QDC_HEAT

Element_name = QDC_Heat , etc…
< heat conduction material data >
< body force data>
< connectivity data >

The element is used for solution of the following scalar heat equation:

$$\rho c \frac{d\theta}{dt} - \nabla \cdot [\underset{\sim}{k} \cdot \nabla \theta] + \left(T_0 \underset{\sim}{\beta} \cdot \nabla \underset{\sim}{v} \right) = \rho h$$

where $\theta =$ temperature, $\underset{\sim}{k} =$ thermal conductivity, $\underset{\sim}{v} =$ solid velocity, $\underset{\sim}{\beta} =$ thermal moduli, $T_0 =$ reference temperature, $\rho =$ mass density, $c =$ specific heat, $h =$ heat source. For heat diffusion (parabolic mode) and heat conduction (elliptic mode) one degree of freedom is assigned to each node for the temperature.

9.2.0.9 Heat Transport Equation

HEAT_TRANSPORT

Element_name = HEAT_TRANSPORT, etc…
< material data >
<body force data >
< connectivity data >

The element is used for the solution of the following scalar equation:

$$\rho c \frac{d\theta}{dt} + \rho c \underset{\sim}{v} \cdot \nabla\theta - \nabla \cdot \left[\underset{\sim}{k} \cdot \nabla\theta \right] + p \nabla \cdot \underset{\sim}{v} - \underset{\sim}{\tau} : \nabla \underset{\sim}{v} = \rho h$$

where θ = temperature; ρ = mass density; c = specific heat; $\underset{\sim}{k}$ = diffusion/dispersion coefficient; $\underset{\sim}{v}$ = flow velocity; p = fluid pressure; $\underset{\sim}{\tau}$ = fluid viscous stress; and h = heat source. One degree of freedom is assigned to each node for the temperature.

9.2.0.10 Electric Charge Equation

QDC_CHARGE

Element_name = QDC_charge, etc...
< charge conduction material model >
< body force data >
< connectivity data >

The element is used for solution of the following electric charge equation:

$$\nabla \cdot \left(\underset{\sim}{k} \cdot \nabla \varphi \right) - \left(\nabla \cdot (\underset{\sim}{e} : \underset{\sim}{\varepsilon}^s) \right) - \rho_e = 0$$

where φ = electric potential, $\underset{\sim}{k}$ = permittivity matrix, ρ_e = electric volume charge density, $\underset{\sim}{\varepsilon}^s$ = solid strain, and $\underset{\sim}{e}$ = piezoelectric constants (third-order tensor, viz., $(\underset{\sim}{e} : \underset{\sim}{\varepsilon}^s)_i = e_{ikl}\varepsilon^s_{kl}$). For this element, one degree of freedom is assigned to each node for the electric potential.

9.2.0.11 Coupled Porous Solid – Pore Fluid Equations

9.2.0.11.1 Diffusive equations:

QDC_POROUS_2

Element_name = QDC_Porous_2 , etc…
< stress material data >
< scalar diffusion material data >
< body force data >
< connectivity data >
< field output data >

In the *parabolic mode* (diffusion analysis) the element is used for solution of the following coupled equations:

$$\nabla \cdot \left[\underset{\sim}{\sigma}'^{s} - p^{w} \underset{\sim}{\delta} \right] + \rho \underset{\sim}{b} = 0$$

$$\varphi C^{w} \frac{dp^{w}}{dt} - \nabla \cdot \left[\frac{\underset{\sim}{k}}{\gamma_{w}} \cdot \left(\nabla p^{w} - \rho_{w} \underset{\sim}{b} \right) \right] + \nabla \cdot \underset{\sim}{v}^{s} = 0$$

where $\underset{\sim}{\sigma}'^{s}$ = solid (effective) stress, $\underset{\sim}{v}^{s}$ = solid velocity, $\underset{\sim}{b}$ = body force (per unit mass), p^{w} = pore fluid pressure, $\rho = \rho^{s} + \rho^{w}$ = total mass density, $\rho^{s} = (1-\varphi)\rho_{s}$ and $\rho^{w} = \varphi \rho_{w}$ with ρ_{s} = solid mass density, ρ^{w} = fluid mass density and φ = porosity; $C^{w} (= 1/\lambda^{w})$ = fluid compressibility, $\underset{\sim}{k}$ = hydraulic conductivity [L/T], and $\gamma_{w} = \rho_{w} g$ = fluid unit weight, $g = (\|\underset{\sim}{b}\|)$ = acceleration of gravity. In that case, NSD solid kinematic degrees of freedom are assigned to each node, in the $x_1, x_2, (and\, x_3)$ directions, respectively, and the degree of freedom (NSD+1) is assigned to the pore fluid pressure.

9.2.0.11.2 Dynamical equations:

QDC_POROUS_1

Element_name = QDC_Porous_1 , etc…
< stress material data >
< scalar diffusion material data >
< body force data >
< connectivity data >
< field output data >

In the *hyperbolic mode* (dynamic analysis) the element is used for solution of the following coupled equations:

$$\rho^s \underset{\sim}{a}^s = \nabla \cdot \underset{\sim}{\sigma}'^s - (1-\varphi)\nabla p^w - \underset{\sim}{\xi} \cdot \left(\underset{\sim}{v}^s - \underset{\sim}{v}^w \right) + \rho^s \underset{\sim}{b}$$

$$\rho^w \frac{d\underset{\sim}{v}^w}{dt} = \rho^w \left(\underset{\sim}{v}^s - \underset{\sim}{v}^w \right) \cdot \nabla \underset{\sim}{v}^w - \varphi \nabla p^w + \underset{\sim}{\xi} \cdot \left(\underset{\sim}{v}^s - \underset{\sim}{v}^w \right) + \rho^w \underset{\sim}{b}$$

where $\underset{\sim}{\sigma}'^s =$ solid (effective) stress, $\underset{\sim}{a}^s =$ solid acceleration, $\underset{\sim}{v}^s (\underset{\sim}{v}^w)$ solid (fluid) velocity, $\underset{\sim}{b} =$ body force (per unit mass), $p^w =$ pore fluid pressure $\rho^s = (1-\varphi)\rho_s$ and $\rho^w = \varphi \rho_w$ with $\rho_s =$ solid mass density, $\rho_w =$ fluid mass density and $\varphi =$ porosity; $\underset{\sim}{\xi} = \varphi^2 \gamma_w \underset{\sim}{k}^{-1}$ with $\underset{\sim}{k} =$ hydraulic conductivity [L/T] and $\gamma_w = \rho_w g =$ fluid unit weight, $g = (\|\underset{\sim}{b}\|) =$ acceleration of gravity. In that case, NSD solid kinematic degrees of freedom are assigned to each node, in the $x_1, x_2, (and\, x_3)$ directions, respectively, and the degrees of freedom (NSD+1), (NSD+2), etc… are assigned to the fluid motion in the $x_1, x_2, (\text{and} x_3)$ directions, respectively.

In the case of a *compressible* pore fluid, the pore fluid pressure is determined from the computed velocities through time integration of the following equation:

$$\mathrm{Dp^w/Dt} = -\left(\lambda^w/\varphi\right)\left[\nabla \cdot (1-\varphi)\underset{\sim}{v}^s + \nabla \cdot \left(\varphi \underset{\sim}{v}^w\right) \right]$$

where $\lambda^w =$ fluid bulk modulus. In the case of an *incompressible* pore fluid, the pore fluid pressure is determined from the computed velocities through the following equation:

$$\mathrm{p^w} = -\left(\lambda^w/\varphi\right)\left[\nabla \cdot (1-\varphi)\underset{\sim}{v}^s + \nabla \cdot \left(\varphi \underset{\sim}{v}^w\right) \right]$$

where $\lambda^w =$ a *penalty* parameter.

References / Bibliography

1. Prevost, J. H.,"Mechanics of Continuous Porous Media," *Int. J. Eng. Sci.*, Vol. 18, No. 5, (1980), pp.787-800.

2. Prevost, J.H., et al.,"Offshore Gravity Structures: Analysis," *J. Geotech. Eng. Div.*, ASCE, Vol. 107, No. GT2, (1981), pp. 143-165.

3. Prevost, J.H.,"Nonlinear Transient Phenomena in Saturated Porous Media," *Comp. Meth. Appl. Mech. Eng.,* Vol. 30, No. 1, (1982), pp. 3-18.4.

4. Prevost, J.H.,"Implicit-Explicit Schemes for Nonlinear Consolidation," *Comp. Meth. Appl. Mech. Eng.*, Vol. 39, (1983), pp. 225-239.

5. Prevost, J.H.,"Nonlinear Transient Phenomena in Soil Media," Chapt. 26 in *Mechanics of Engineering Materials*, Eds C.S. Desai and R.H. Gallagher, Wiley, (1984), pp. 515-533.

6. Prevost, J.H.,"Wave Propagation in Fluid-Saturated Porous Media: an Efficient Finite Element Procedure," *Int. J. Soil Dyn. Earthq. Eng.*, Vol. 4, No. 4, (1985), pp. 183-202.

9.2.0.12 Darcy Flow Equation

Darcy's law for the flow of viscous fluid in a permeable medium, and conservation of mass are written as follows:

$$\underset{\sim}{\bar{q}} = -\frac{\underset{\sim}{k}}{\mu} \cdot (\nabla p - \rho \underset{\sim}{b}) \qquad (1)$$

$$\nabla \cdot \underset{\sim}{\bar{q}} = 0 \qquad (2)$$

where $\underset{\sim}{\bar{q}} =$ Darcy velocity; p = fluid pressure, $\underset{\sim}{b}$ = body force (per unit mass), ρ = fluid mass density, $\underset{\sim}{k}$ = intrinsic permeability with units [L^2] (see Section 10.16), and μ = fluid viscosity.

Two formulations of the Darcy flow equation are available: (1) a primal formulation which amounts to solving a Poisson problem for the pressure; and (2) a mixed formulation for which both the pressure and velocity are treated as unknowns.

9.2.0.12.1 Pressure Formulation:

The equation is simply obtained by taking the divergence of Eq. 1, and leads to:

$$\nabla \cdot \left[\frac{\underset{\sim}{k}}{\mu} \cdot (\nabla p - \rho \underset{\sim}{b}) \right] = 0$$

The equation is often written in terms of the total head h as:

$$\nabla \cdot \left[\gamma_w \frac{\underset{\sim}{k}}{\mu} \cdot \nabla h \right] = 0$$

where $\gamma_w \frac{\underset{\sim}{k}}{\mu}$ = hydraulic conductivity with units [L/T]; $h = h_p + h_e$, with $h_p = p/\gamma$ = pressure head, and h_e = elevation head defined such that: $\nabla h_e = -\underset{\sim}{b}/g = -\underset{\sim}{b}/\|\underset{\sim}{b}\|$, viz., $h = h_p + x_2$ for the case x_2 vertical and oriented positively upward.

There is one degree of freedom assigned to each node for the fluid pressure.

QDC_DARCY_PRESSURE

Element_name = QDC_Darcy_pressure , etc…
< scalar diffusion material data >
< body force data >
< connectivity data >

9.2.0.12.2 Mixed Formulation:

In the mixed formulation, both Eqs. 1 and 2 are used to compute the fluid pressure and velocity vector components. A stabilized mixed formulation is used. The element may be used for both 2D and 3D applications, and equal order interpolations are used for the fluid pressure and velocity. There are (NSD+1) degrees of freedom assigned to each node for the velocity vector components and the fluid pressure.

QDC_DARCY_MIXED

Element_name = QDC_Darcy_mixed , etc…
< scalar diffusion material data >
< body force data >
< connectivity data >

References / Bibliography

1. Masud, A. and T.J.R. Hughes, "A Stabilized Mixed Finite Element Method for Darcy Flow," *Comp. Meth. Appl. Mech. Eng.*, Vol. 191, (2002), pp. 4341-4370.

9.2.0.13 Pressure Diffusion Equation

The element is used for solution of the following equations:

$$\underset{\sim}{\bar{q}}_f = -\frac{\underset{\sim}{k}}{\mu}\cdot\left(\nabla p_f - \rho_f \underset{\sim}{b}\right) \qquad (1)$$

$$\left[\frac{1}{N} + \phi C_f\right]\frac{dp_f}{dt} + \nabla\cdot\underset{\sim}{\bar{q}}_f + b\nabla\cdot\underset{\sim}{v}^s = 0 \qquad (2)$$

$$b = 1 - \frac{K_m}{K_s} \qquad \frac{1}{N} = \frac{b-\phi_0}{K_s} \qquad C_f = \frac{1}{\rho_f}\frac{\partial \rho_f}{\partial p_f} = \textit{fluid compressibility}$$

where $\underset{\sim}{\bar{q}}_f$ = Darcy fluid velocity, p_f = fluid pressure, $\underset{\sim}{k}$ = permeability (intrinsic; units [L^2]), μ = fluid viscosity, ρ_f = fluid mass density, $\underset{\sim}{v}^s$ = solid velocity, C_f = fluid compressibility, and ϕ = porosity.

Two formulations are available: (1) a primal formulation which solves a pressure equation, and (2) a mixed formulation for which both the pressure and fluid motion are treated as unknowns.

9.2.0.13.1 Pressure Formulation:

The equation is as follows:

$$\left[\frac{1}{N} + \phi C_f\right]\frac{dp_f}{dt} - \nabla\cdot\left[\frac{\underset{\sim}{k}}{\mu}\cdot\left(\nabla p_f - \rho_f \underset{\sim}{b}\right)\right] + b\nabla\cdot\underset{\sim}{v}^s = 0$$

There is one degree of freedom assigned to each node for the pressure.

QDCD_PRESSURE

Element_name = QDCD_PRESSURE , etc…
< scalar diffusion material data >
< body force data >
< connectivity data >

9.2.0.13.2 Mixed Formulation:

There are (NSD+1) degrees of freedom assigned to each node for the fluid velocity and the fluid pressure.

QDCD _MIXED

Element_name = QDCD_MIXED, etc…
< scalar diffusion material data >
< body force data >
< connectivity data >

Notes . .

9.2.0.14 Scalar Transport Equation in Incompressible Miscible Multi-Phase Flows

cmi_QDCA

Element_name = cmi_QDCA, etc…
< scalar diffusion material data >
< body force data >
< connectivity data >

The element is used for solution of the following scalar mass balance equation in miscible incompressible flows:

$$\varphi \frac{d}{dt} c + \nabla \cdot \left[c \, \underset{\sim}{\bar{q}}_f - \underset{\sim}{D} \cdot \nabla c \right] + c \left[\frac{1}{J} \frac{d\phi}{dt} \right] = \bar{c} \, Q$$

$$\frac{d\phi}{dt} = b \nabla \cdot \underset{\sim}{v}^s + \frac{1}{N} \frac{dp_f}{dt} - 3\alpha_s \left(b - \phi_0 \right) dT \qquad \frac{1}{N} = \frac{b - \phi_0}{K_s} \qquad b = 1 - \frac{K}{K_s}$$

where $c\left(\underset{\sim}{x}, t\right)$ = volumetric concentration of the invading fluid; $\underset{\sim}{\bar{q}}_f\left(\underset{\sim}{x}, t\right)$ = velocity of the fluid mixture (typically obtained by solving Darcy's equation per Section 9.2.0.12); $\underset{\sim}{D}\left(\underset{\sim}{x}\right)$ = diffusion-dispersion tensor; $\varphi\left(\underset{\sim}{x}\right)$ = porosity of the medium; $Q\left(\underset{\sim}{x}, t\right)$ = injection/extraction volumetric flow rate; $\bar{c}\left(\underset{\sim}{x}, t\right)$ is either the specified concentration of the injected fluid at injection wells or $\bar{c}\left(\underset{\sim}{x}, t\right) = c\left(\underset{\sim}{x}, t\right)$ is the resident concentration at production wells. One degree of freedom is assigned to each node for the concentration.

9.2.0.15 Immiscible Multi-Phase Flow Equation

QDC_MFLOW

Element_name = QDC_mflow, etc…
< scalar diffusion material data >
< body force data >
< connectivity data >
< field output data >

The element is used for solution of the following mass balance equations in multi-phase fluid flow thru porous media:

$$\frac{\partial}{\partial t}\left(\varphi\,\rho_\beta S_\beta\right) \;+\; \nabla\cdot\left[\rho_\beta\,\underline{\underline{\bar{q}}}_\beta\right] \;=\; Q_\beta$$

where

$$\underline{\underline{\bar{q}}}_\beta = \varphi\,S_\beta\,\underline{v}^\beta = -\,\frac{k_{r\beta}}{\mu_\beta}\,\underline{\underline{k}}\cdot\left[\nabla p_\beta - \rho_\beta\,\underline{b}\right] + \varphi\,S_\beta\,\underline{v}^s$$

$$\varphi\,S_\beta\left(\underline{v}^\beta - \underline{v}^s\right) = -\frac{k_{r\beta}}{\mu_\beta}\,\underline{\underline{k}}\cdot\left[\nabla p_\beta - \rho_\beta\,\underline{b}\right]$$

β = fluid phase number (β =1, …, number_of_phases)
$\underline{q}_\beta$ = volumetric flux of phase β (volume per total area per unit time)
$\underline{v}^\beta$ = fluid (seepage) velocity of phase β
$\underline{v}^s$ = solid velocity
φ = porosity (volume of voids per unit volume)
S_β = degree of saturation (ratio volume of fluid to volume of voids); $\sum_\beta S_\beta = 1$
ρ_β = fluid phase mass density
μ_β = fluid viscosity
$k_{r\beta}$ = relative permeability
$\underline{\underline{k}}$ = intrinsic absolute permeability (units [L^2])
p_β = fluid pressure
$\underline{b}$ = body force (per unit mass)
Q_β = source (or sink) of mass to phase β (units: [$M/L^3/T$]) = $\sum_j Q_\beta^j$

One degree of freedom is assigned to the fluid pressure for each phase.

For two-phase flows $(\beta = 1, 2)$, the two coupled equations can be written as follows:

$$\varphi \rho_\beta S_\beta C_\beta \frac{dp_\beta}{dt} + \varphi \rho_\beta \frac{dS_\beta}{dt}$$
$$-\nabla \cdot \left[\rho_\beta \frac{\underset{\sim}{k}_\beta}{\mu_\beta} \cdot \left(\nabla p_\beta - \rho_\beta \underset{\sim}{b} \right) \right] + \rho_\beta S_\beta \nabla \cdot \underset{\sim}{v}^s = Q_\beta$$

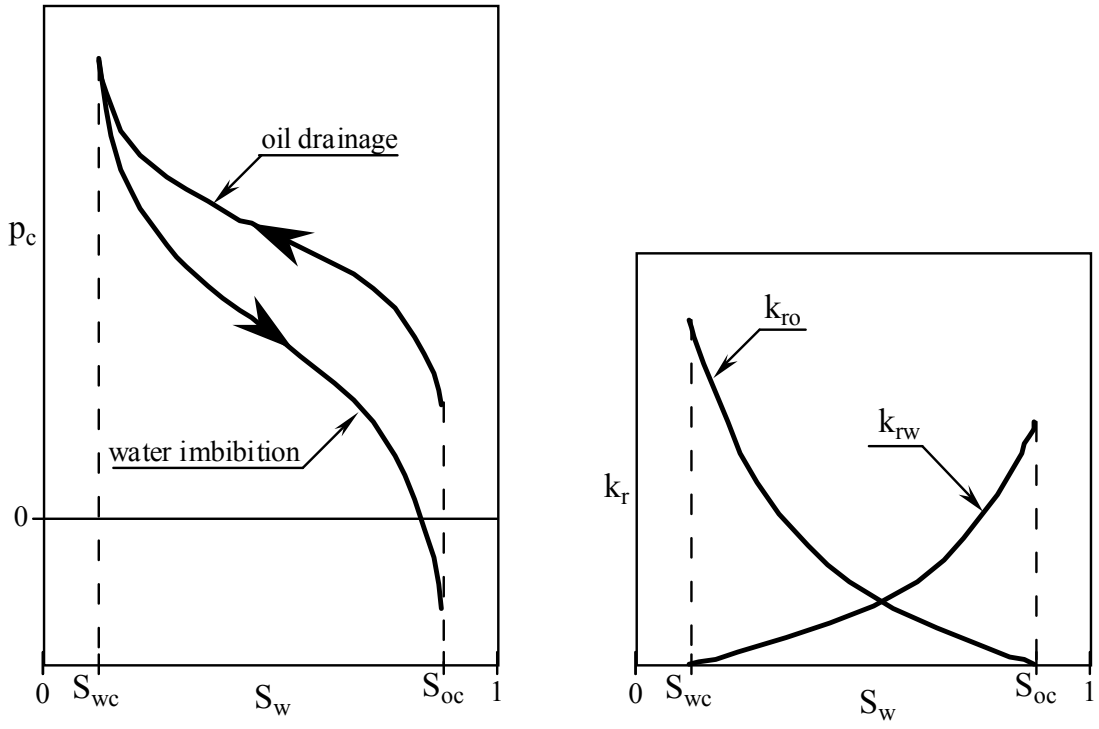
where

C_β = fluid compressibility $\left(C_\beta = \frac{1}{\rho_\beta} \frac{\partial \rho_\beta}{\partial \mathrm{p}_\beta} \right)$

$\underset{\sim}{k}_\beta = k_{r\beta}(S_w)\, \underset{\sim}{k}$ = total permeability (units [L^2])

$\frac{d\bullet}{dt} = \frac{\partial \bullet}{\partial t} + \underset{\sim}{v}^s \cdot \nabla(\bullet)$ = material derivative

The degree of water saturation $S_w = S_w(P_c)$ is a nonlinear (multi-valued) function of $P_c = p_o - p_w =$ capillary pressure, shown schematically in Figure 9.2.0.15a, where $S_{wc} =$ residual water saturation and $S_{wro} = 1 - S_{or}$ with $S_{or} =$ residual oil saturation. The relative permeability coefficient $k_{ri} = k_{ri}(S_w)$ for each fluid phase is a nonlinear function of the water saturation shown schematically in Figure 9.2.0.15b (see Section 10.15 for further details).



(a) Typical $S_w(p_c)$ Curves (b) Typical k_{ri} Curves

Figure 9.2.0.15.1 Typical Material Curves

9.2.0.16 Pressure Equation in Incompressible Immiscible Multi-Phase Flows

The element is used for solution of the following pressure equation in incompressible immiscible multi-phase flows:

$$\nabla\bullet\bar{\underset{\sim}{q}} + \frac{1}{J}\frac{d\phi}{dt} = \sum_{\beta}\frac{Q_\beta}{\rho_\beta} \qquad \phi = J\varphi = Lagrangian\ porosity$$

where the total flux:

$$\bar{\underset{\sim}{q}} = \sum_{\beta}\bar{\underset{\sim}{q}}_\beta = -\left(\sum_{\beta}k_\beta\right)\underset{\sim}{k}\bullet\left[\nabla P - \bar{\rho}\underset{\sim}{b}\right] \qquad \bar{\rho} = \frac{\sum_{\beta}\rho_\beta k_\beta}{\sum_{\beta}k_\beta} \qquad k_\beta = \frac{k_{r\beta}}{\mu_\beta} = mobility$$

and:

$$\frac{d\phi}{dt} = b\nabla\bullet\underset{\sim}{v}^s + \frac{1}{N}\frac{dP}{dt} - 3\alpha_s\left(b-\phi_0\right)dT \qquad \frac{1}{N} = \frac{b-\phi_0}{K_s} \qquad b = 1-\frac{K}{K_s}$$

and the global pressure P (Chavent [1981, 1984]) for $n_p = 2$ phases:

$$P = \frac{1}{2}(p_1+p_2) + \gamma(S_L) \qquad \gamma(S_L) = \int_{S_{L_s}}^{S_L}\frac{1}{2}\frac{k_2-k_1}{k_1+k_2}\frac{dP_c}{dS_L}dS_L$$

where

$$P_c = p_2 - p_1 = P_c(S_L) = capillary\ \ pressure$$

The phase fluxes can be computed as:

$$\bar{\underset{\sim}{q}}_\alpha = \frac{k_\alpha}{\sum_{\beta}k_\beta}\bar{\underset{\sim}{q}} + \frac{k_\alpha}{\sum_{\beta}k_\beta}\underset{\sim}{k}\bullet\left[\sum_{\beta\neq\alpha}k_\beta\left[\nabla P_c^{\alpha\beta} + \left(\rho_\alpha-\rho_\beta\right)\underset{\sim}{b}\right]\right] \qquad P_c^{\alpha\beta} = p_\beta - p_\alpha = capillary\ pressure$$

β = fluid phase number (β=1, …, number_of_phases)
$\bar{\underset{\sim}{q}}_\beta$ = volumetric flux of phase β (volume per total area per unit time)
$\phi = J\varphi$ = Lagrangian porosity (volume of voids per unit volume)
φ = Eulerian porosity

S_β = degree of saturation (ratio volume of fluid to volume of voids); $\sum_\beta S_\beta = 1$

ρ_β = fluid phase mass density

μ_β = fluid viscosity

$k_{r\beta}$ = relative permeability

$\underset{\sim}{k}$ = intrinsic absolute permeability (units [L^2])

p_β = fluid pressure

$\underset{\sim}{b}$ = body force (per unit mass)

Q_β = source (or sink) of mass to phase β (units: [$M/L^3/T$]) = $\sum_j Q_\beta^j$

$\frac{d\bullet}{dt} = \frac{\partial \bullet}{\partial t} + \underset{\sim}{v}^s \bullet \nabla(\bullet)$ = material derivative

9.2.0.16.1 Pressure Formulation:

There is one degree of freedom assigned to the global pressure P.

QDCP_PRESSURE

Element_name = QDCP_PRESSURE, etc…
< scalar diffusion material data >
< body force data >
< connectivity data >

9.2.0.16.2 Mixed Formulation:

In the mixed formulation both the global pressure P and the total flux $\underset{\sim}{q}$ are treated as unknowns.

QDCP_MIXED

Element_name = QDCP_MIXED, etc…
< scalar diffusion material data >
< body force data >
< connectivity data >

9.2.0.17 Saturation Equation in Incompressible Immiscible Multi-Phase Flows

cmi_QDCS

Element_name = cmi_QDCS, etc…
< scalar diffusion material data >
< body force data >
< connectivity data >

The element is used for solution of the saturation equation in incompressible immiscible multi-phase flows:

$$\varphi\frac{dS_\alpha}{dt}+\nabla\bullet\left[\frac{k_\alpha}{\sum_\beta k_\beta}\underset{\sim}{\overline{q}}+\frac{k_\alpha}{\sum_\beta k_\beta}\underset{\sim}{k}\bullet\left[\sum_{\beta\neq\alpha}k_\beta\left[\nabla P_c^{\alpha\beta}+(\rho_\alpha-\rho_\beta)\underset{\sim}{b}\right]\right]\right]+S_\alpha\left[\frac{1}{J}\frac{d\phi}{dt}\right]=\frac{Q_\alpha}{\rho_\alpha}$$

where $\overline{q}$ = total flux (typically obtained by solving the appropriate pressure equation; see Section 9.2.0.16).

β = fluid phase number (β =1, …, number_of_phases)
$\overline{q}_\beta$ = volumetric flux of phase β (volume per total area per unit time)
$\phi = J\varphi =$ Lagrangian porosity
φ = Eulerian porosity (volume of voids per unit volume)
S_β = degree of saturation (ratio volume of fluid to volume of voids); $\sum_\beta S_\beta = 1$
ρ_β = fluid phase mass density
μ_β = fluid viscosity
$k_{r\beta}$ = relative permeability
$\underset{\sim}{k}$ = intrinsic absolute permeability (units [L^2])
$k_\beta = \frac{k_{r\beta}}{\mu_\beta}$ = mobility
p_β = fluid pressure
$\underset{\sim}{b}$ = body force (per unit mass)
Q_β = source (or sink) of mass to phase β (units: [$M/L^3/T$]) = $\sum_j Q_\beta^j$
$\frac{d\bullet}{dt}=\frac{\partial\bullet}{\partial t}+\underset{\sim}{v}^s\bullet\nabla(\bullet)$ = material derivative

One degree of freedom is assigned to each node for the saturation.

9.2.0.18 Pressure Equation in Compressible Immiscible Compositional Multi-Phase Flows

The element is used for solution of the pressure equation in compressible immiscible compositional multi-phase flows:

$$\varphi \frac{d}{dt}\left(\sum_{\beta} \rho_\beta S_\beta \right) + \nabla \bullet \underset{\sim}{q} + \sum_{\beta} \rho_\beta S_\beta \left[\frac{1}{J} \frac{d\phi}{dt} \right] = \sum_{\beta} Q_\beta$$

$$\frac{d\phi}{dt} = b \nabla \bullet \underset{\sim}{v}^s + \frac{1}{N} \frac{dP}{dt} - 3\alpha_s \left(b - \phi_0 \right) dT \qquad \frac{1}{N} = \frac{b - \phi_0}{K_s} \qquad b = 1 - \frac{K}{K_s}$$

where the total mass flux:

$$\underset{\sim}{q} = \sum_{\beta} \rho_\beta \underset{\sim}{\bar{q}}_\beta = -\left(\sum_{\beta} \rho_\beta k_\beta \right) \underset{\sim}{k} \bullet \left[\nabla P - \bar{\rho} \underset{\sim}{b} \right] \qquad \bar{\rho} = \frac{\sum_{\beta} \rho_\beta^2 k_\beta}{\sum_{\beta} \rho_\beta k_\beta} \qquad k_\beta = \frac{k_{r\beta}}{\mu_\beta} = mobility$$

and the global pressure *P* (Chavent [1986]) for $n_p = 2$ phases:

$$P = \frac{1}{2}(p_1 + p_2) + \gamma(S_L) \qquad \gamma(S_L) = \int_{S_{L_c}}^{S_L} \frac{1}{2} \frac{\rho_2 k_2 - \rho_1 k_1}{\rho_1 k_1 + \rho_2 k_2} \frac{dP_c}{dS_L} dS_L$$

where

$$P_c = p_2 - p_1 = P_c(S_L) = capillary\ pressure$$

The phase mass fluxes can be computed as:

$$\underset{\sim}{q}_\alpha = \rho_\alpha \underset{\sim}{\bar{q}}_\alpha = \frac{\rho_\alpha k_\alpha}{\sum_{\beta} \rho_\beta k_\beta} \underset{\sim}{q} + \frac{\rho_\alpha k_\alpha}{\sum_{\beta} \rho_\beta k_\beta} \underset{\sim}{k} \bullet \left[\sum_{\beta \neq \alpha} \rho_\beta k_\beta \left[\nabla P_c^{\alpha\beta} + (\rho_\alpha - \rho_\beta) \underset{\sim}{b} \right] \right]$$

β = fluid phase number (β=1, …, number_of_phases)
$\bar{q}_\beta$ = volumetric flux of phase β (volume per total area per unit time)
$\phi = J\varphi$ = Lagrangian porosity
φ = Eulerian porosity (volume of voids per unit volume)
S_β = degree of saturation (ratio volume of fluid to volume of voids); $\sum_{\beta} S_\beta = 1$
ρ_β = fluid phase mass density

μ_β = fluid viscosity

$k_{r\beta}$ = relative permeability

$\underset{\sim}{k}$ = intrinsic absolute permeability (units [L^2])

$k_\beta = \dfrac{k_{r\beta}}{\mu_\beta}$ = mobility

p_β = fluid pressure

$\underset{\sim}{b}$ = body force (per unit mass)

Q_β = source (or sink) of mass to phase β (units: [$M/L^3/T$]) = $\sum_j Q_\beta^j$

$\dfrac{d\bullet}{dt} = \dfrac{\partial \bullet}{\partial t} + \underset{\sim}{v}^s \bullet \nabla(\bullet)$ = material derivative

Two formulations of the equation are available: (1) a primal formulation for which the total pressure is the unknown; and (2) a mixed formulation for which both the total pressure and the total mass flux are treated as unknowns.

9.2.0.18.1 Pressure Formulation:

The equation is obtained by substituting the total mass flux in the conservation equation. There is one degree of freedom assigned to each node for the total pressure.

cmi_QDCP_PRESSURE

Element_name = cmi_QDCP_pressure, etc…
< scalar diffusion material data >
< body force data >
< connectivity data >

9.2.0.18.2 Mixed Formulation:

Both the total pressure and the mass flux are treated as unknowns. There are (NSD+1) degrees of freedom assigned to each node for the total mass flux and the total pressure.

cmi_QDCP_MIXED

Element_name = cmi_QDCP_mixed, etc…
< scalar diffusion material data >
< body force data >
< connectivity data >

9.2.0.19 Mole/Mass Transport Equation in Compressible Immiscible Compositional Multi-Phase Flows

The element is used for solution of the following mass-balance equation(s) in compressible immiscible compositional multi-phase flows:

$$\varphi\frac{d}{dt}\left[Z^k\sum\rho_\beta S_\beta\right]+\nabla\bullet\left[\sum_\beta\rho_\beta X_\beta^k\bar{\underset{\sim}{q}}_\beta\right]-\nabla\bullet\left[\varphi\sum_\beta\rho_\beta S_\beta\mathbf{D}_\beta^k\bullet\nabla X_\beta^k\right]+Z^k\sum_\beta\rho_\beta S_\beta\left[\frac{1}{J}\frac{d\phi}{dt}\right]=Q^k$$

where $\bar{\underset{\sim}{q}}_\beta$ = flux of phase β (typically obtained by solving the appropriate pressure equation; see Section 9.2.0.18);

$Z^k=\dfrac{\sum_\beta\rho_\beta S_\beta X_\beta^k}{\sum_\beta\rho_\beta S_\beta}$ = mole/mass fraction of component *k* in the mixture $\left(\sum_k Z^k=1\right)$

X_β^k = mole/mass fraction of component *k* in phase $\beta\left(\sum_k X_\beta^k=1\right)$

$m^k=\varphi\sum_\beta\rho_\beta S_\beta X_\beta^k=\varphi Z^k\sum_\beta\rho_\beta S_\beta$ = mass of component *k*

β = fluid phase number (β =1, …, number_of_phases)

$\bar{\underset{\sim}{q}}_\beta$ = volumetric flux of phase β (volume per total area per unit time)

$\phi=J\varphi$ = Lagrangian porosity

φ = Eulerian porosity (volume of voids per unit volume)

S_β = degree of saturation (ratio volume of fluid to volume of voids); $\sum_\beta S_\beta=1$

ρ_β = fluid phase mass density

$\mathbf{D}_\beta^k$ = diffusion/dispersion coefficient matrix (units {L^2/T]

$\underset{\sim}{b}$ = body force (per unit mass)

Q_β = source (or sink) of mass to phase β (units: [M/L^3/T]) = $\sum_j Q_\beta^j$

Q^k = source (or sink) to component $k=\sum_\beta Q_\beta^k$ = (units [M/L^3/T])

$\dfrac{d\bullet}{dt}=\dfrac{\partial\bullet}{\partial t}+\mathbf{v}^s\bullet\nabla(\bullet)$ = material derivative

One degree of freedom is assigned to each node for the mole or mass fraction Z^k for which the equation is solved.

cmi_QDCZ

Element_name = cmi_QDCZ, etc…
< scalar diffusion material data >
< body force data >
< connectivity data >

Notes . .

9.2.0.20 Multi-Phase Heat Transfer Equation

MULTI_PHASE_HEAT

Element_name = Multi_phase_heat, etc…
< heat conduction material data >
< body force data >
< connectivity data >

The element is used for solution of the following energy balance equation in multi-phase fluid flow thru porous media:

$$\frac{dU_t}{dt} + \sum_{\beta} \rho_\beta \bar{\underset{\sim}{q}}_\beta \bullet \nabla h_\beta \; - \nabla \bullet \left[\underset{\sim}{K}_T \nabla T \right] = \underset{\sim}{\sigma} : \underset{\sim}{d}^S + \underset{\sim}{b} \bullet \left[\sum_{\beta} \rho_\beta \bar{\underset{\sim}{q}}_\beta \right]$$

where

$$U_t = H_t - P \quad [energy/V] \qquad H_t = (J - \phi)\rho_R c_R (T - T_0) + \phi \sum_{\beta} \rho_\beta S_\beta h_\beta$$

$$\bar{\underset{\sim}{q}}_\beta = -\frac{k_{r\beta}}{\mu_\beta} \underset{\sim}{k} \bullet \left[\nabla p_\beta - p_\beta \underset{\sim}{b} \right]$$

β = fluid phase number (β =1, …, number_of_phases)
T = temperature (absolute)
$\underset{\sim}{K}_T$ = total thermal conductivity $\left[W/(m^\circ K) = J/(sm^\circ K) \right]$
$\phi = J\varphi$ = Lagrangian porosity
φ = Eulerian porosity (volume of voids per unit volume)
S_β = degree of saturation (ratio volume of fluid to volume of voids); $\sum_{\beta} S_\beta = 1$
μ_β = fluid viscosity
$k_{r\beta}$ = relative permeability
$\underset{\sim}{k}$ = intrinsic absolute permeability (units [L^2])
p_β = fluid pressure
ρ_R, ρ_β = rock and fluid phase mass densities, respectively
$\underset{\sim}{b}$ = body force (per unit mass)
$u_\beta = c_\beta^V T$ = specific internal energy
c_R = rock specific heat capacity $\left[J/(kg^\circ C) \right]$

$$h_\beta = \left(u_\beta + \frac{p_\beta}{\rho_\beta} \right) = c_\beta^P T = \text{fluid specific enthalpy} \left[J / kg \right]$$

c_β^V = fluid heat capacity at constant volume

c_β^P = fluid heat capacity at constant pressure

Q = heat source

One degree of freedom is assigned to each node for the temperature.

9.2.0.21 Poro Heat Equation

PORO_ HEAT

Element_name = Poro_heat, etc…
< heat conduction material data >
< body force data >
< connectivity data >

The element is used for solution of the following energy balance equation in porous media:

$$\left[(1-\varphi)\rho_R c_R + \varphi\rho_f c_f\right]\frac{dT}{dt} + \rho_f c_f T\ \underset{\sim}{\bar{q}}_f \bullet \nabla T - \nabla\bullet\left[\underset{\sim}{K}_T \bullet \nabla T\right] + 3\alpha_S K^S T_0 \nabla \bullet \underset{\sim}{v}^S - 3\alpha_m T_0 \frac{dp_f}{dt} = 0$$

where

$\underset{\sim}{K}_T$ = thermal conductivity $\left[W/(m\,°C) = J/(\sec m\,°C)\right]$
$c_R\,(c_f)$ = rock (fluid) specific heat capacity $\left[J/(kg\,°C)\right]$
$\rho_R\,(\rho_f)$ = rock (fluid) mass density $\left[kg/m^3\right]$

$$\alpha_m = (b-\varphi_0)\alpha_S + \varphi\alpha_f \qquad \alpha_f = \sum_\beta \alpha_\beta S_\beta \qquad 3\alpha_\beta = -\frac{1}{\rho_\beta}\frac{\partial\rho_\beta}{\partial T}\Big|_{p_f\ fixed}$$

$$\underset{\sim}{\bar{q}}_f = -\frac{1}{\mu_f}\underset{\sim}{k}\bullet\left[\nabla p_f - \rho_f \underset{\sim}{b}\right] = \left[L/T\right] = \left[e.g.,\ m/\sec\right] \quad \text{(Darcy's Law)}$$

$$\textit{note: in multiphase flows} \quad \rho_f c_f = \sum_\beta \rho_\beta c_\beta S_\beta \qquad \rho_f c_f \underset{\sim}{\bar{q}}_f = \sum_\beta \rho_\beta c_\beta S_\beta \underset{\sim}{\bar{q}}_\beta$$

$$\textit{compressible flows} \quad \underset{\sim}{\bar{q}}_\alpha = \frac{k_\alpha}{\sum_\beta \rho_\beta k_\beta}\left[\underset{\sim}{q} + \underset{\sim}{k}\bullet\left[\sum_{\beta\neq\alpha}\rho_\beta k_\beta\left[\nabla P_C^{\alpha\beta} + (\rho_\alpha - \rho_\beta)\underset{\sim}{b}\right]\right]\right]$$

T = temperature
$\phi = J\varphi$ = Lagrangian porosity
φ = Eulerian porosity (volume of voids per unit volume)
μ_f = fluid viscosity
$\underset{\sim}{k}$ = intrinsic absolute permeability (units [L^2])
p_f = fluid pressure
ρ_R, ρ_f = rock and fluid phase mass densities, respectively
$\underset{\sim}{b}$ = body force (per unit mass)

One degree of freedom is assigned to each node for the temperature.

9.2.0.22 Laplace Equation

QDC_LAPLACE

Element_name = QDC_Laplace, etc…
< connectivity data >

The element is used for the solution of the following elliptic equation:

$$\nabla^2 \varphi = 0$$

where $\phi = \phi(\underset{\sim}{x})$ = potential. One degree of freedom is assigned to each node for the potential.

9.2.0.23 Stream Function Equation

QDC_STREAM

Element_name = QDC_stream, etc…
< connectivity data >

The element is used for the solution of the stream function equation:

$$\nabla^2 \psi = 0$$

where $\psi = \psi(\underset{\sim}{x})$ stream function, and

$$\frac{\partial \psi}{\partial x} = -\frac{\partial \varphi}{\partial y} \qquad \frac{\partial \psi}{\partial y} = +\frac{\partial \varphi}{\partial x}$$

where $\phi = \phi(\underset{\sim}{x})$ potential. The implementation is restricted to 2-dimension. One degree of freedom is assigned to each node for the stream function.

9.2.0.24 Level Set Equation

QDC_LEVEL

Element_name = QDC_level, etc...
< connectivity data >

The element is used for solution of the level set function equation:

$$\phi_{,t} + \underset{\sim}{v} \bullet \nabla \phi = 0$$

where $\varphi = \varphi(x,t) = \text{level set}$, and $\underset{\sim}{v} = \text{velocity of the level set}$. In the current implementation the velocity is assumed to be normal to the level set (viz., $\underset{\sim}{v} = \mathrm{V}\, \nabla\phi / |\nabla\phi|$). The normal velocity magnitude V is defined on the zero level set curve where $\varphi(x,t) = 0$.

Notes . .

Notes . .

9.2.1 Element Control Information

Note	Variable Name	Type	Default	Description
	Element_name	list	[none]	Element name
	Element_shape eight_node_brick twenty_node_brick six_node_wedge fifteen_node_wedge four_node_tetra ten_node_tetra four_node_quad eight_node_quad nine_node_quad three_node_tri six_node_tri two_node_line three_node_line	list	[none]	Element shape (see Appendix D)
	Analysis_type one_dimensional **plane** axisymmetric three_dimensional	list	[plane]	Analysis type: One-dimensional Two-dimensional / plane strain Axisymmetric Three-dimensional
(1)	Finite_deformation on / **off**	list	[off]	Finite deformation option
	Large_strains on / **off**	list	[off]	Large strains option
	Numerical_integration reduced **full** extended over_extended over_over_extended	list	[full]	Numerical integration option: If reduced: one-point Gaussian quadrature
	Strain_displacement **standard** bbar_select bbar_mean	list	[*]	Strain-displacement option: Standard formulation Selective-reduced integration Mean-dilatational formulation
(2)	Incompatible_modes on / **off**	list	[off]	Incompatible modes option
	Mass_type **lumped** / consistent	list	[lumped]	Mass type If lumped: row-sum technique

(cont'd)

(cont'd)

Note	Variable Name	Type	Default	Description
(3)	Free_field on / **off**	list	[off]	Free field option
(4)	Upwinding_type **off** optimal asymptotic critical	list	[off]	Upwinding option: No upwinding Optimal rule Doubly asymptotic approximation Critical approximation
(4)	Upwinding_viscosity **spatial** temporal	list	[*]	Artificial viscosity for upwinding: Spatial criterion Temporal criterion
	Number_of_stress_points	integer	[*]	Number of stress points option: = 1, one stress point Default: one per integration point
	Implicit_explicit_type implicit explicit **implicit_explicit**	list	[*]	Implicit/explicit option: Implicit element Explicit element Implicit-explicit element
	Eigen_implicit_explicit_type implicit explicit **implicit_explicit**	list	[*]	Eigenvalue solution implicit/explicit option: Implicit element Explicit element Implicit-explicit element
(5)	Fluid_type **incompressible** compressible	list	[*]	Fluid option: Incompressible Compressible
(6)	Stabilization **on** / off	list	[on]	Stabilization option
(7)	Excess_pore_fluid	integer	[0]	Pore fluid pressure reference option: = 0, Total pressure = 1, Excess pressure w.r.t. hydrostatic
	Fluid_cavitation on / **off**	list	[off]	Fluid cavitation option
	Number_of_geometric_sets	integer	[0]	Number of geometric sets
	Body_force_load_time	integer	[0]	Body force load-time function number
	Temperature_load_time	integer	[0]	Temperature load time function number

(cont'd)

(cont'd)

Note	Variable Name	Type	Default	Description
(8)	Permeability_load_time	integer	[0]	Permeability load-time function number
(9)	Fluid_bulk_load_time	integer	[0]	Fluid bulk load-time function number
(10)	Xfem_option on / **off**	list	[off]	Xfem option
(10)	Link_to_crack element(s)	list	[none]	Link to crack element groups; name(s) must be enclosed in quotation marks
(10)	Link_to_grain_boundary	list	[none]	Link to grain boundary; name(s) must be enclosed in quotation marks
	Link_to_level_set	list	[none]	Link to level set groups; name(s) must be enclosed in quotation marks
(11)	Coupling **on** / off	list	[on]	Coupling solid-fluid option
(11)	Porosity_update on / **off**	list	[off]	Porosity update option
(11)	Component_number	integer	[1]	Component number
(11)	Component_name CO2 NaCl CH4 H2S H2O	string	[none]	Component name; name(s) must be enclosed in quotation marks.
	eos_integration **reduced** full	list	[*]	eos numerical integration
(11)	Number_of_wells	integer	[0]	Number of wells, Nwells
(11)	Mass_fraction **on** / off	list	[on]	Mass fraction option (cmi_QDCZ)
(11)	Mole_fraction on / **off**	list	[off]	Mole fraction option (cmi_QDCZ)
(12)	Fluid_cell_pressures on / **off**	list	[off]	Fluid cell pressures option
(12)	Number_of_pressure_load_cases	integer	[0]	Number of pressure load cases

EXAMPLE

```
Element_Group  /
    name = "Group 1"  /
    element_type = continuum  /
    element_shape = four_node_quad  /
    number_of_material_sets = 1  /
    element_name = QDC_Porous  /
        analysis_type = axisymmetric  /
        strain_displacement = bbar_mean  /
        number_of_output_sets = 1

Stress_Model  /
    material_type = linear  /
    material_name = linear_elastic
        material_set_number = 1  /
        youngs_modulus = 1.0E4  /
        poissons_ratio = 0.25  /
        solid_mass_density = 2.0  /
        fluid_mass_density = 1.0  /
        porosity = 0.30

Scalar_Diffusion_Model  /
    material_type = linear  /
    material_name = scalar_diffusion
        material_set_number = 1  /
        mass_density = 1.00E+00  /
        compressiblity = 1.0E-06  /
        permeability  /
            type = isotropic  /
            k_11 = 1.60E-03

Body_force      b_x1=0.00       b_x2=0.00       h=0.00

Field_output
      58          0          17

Nodal_connectivity      etc...
```

Notes/

(1) If finite_deformation = on, output strains are Lagrangian. Otherwise, output strains are "infinitesimal".

(2) Only applicable to four_node_quad and eight_node_brick elements.

(3) This option requires that inside and outside free_field nodes data be defined (see Section 7.5.4).

(4) Applicable to ALE and/or fluid cases.

(5) Only applicable to fluid and/or porous media cases.

(6) Applicable to incompressible or near-incompressible mixed formulations.

(7) Only applicable to porous media.

(8) Only applicable to porous media. Allows a variable permeability to be prescribed.

(9) Allows a variable fluid bulk modulus to be prescribed.

(10) Only applicable to Xfem procedures.

(11) Only applicable to multi-phase fluid flow problems.

(12) Only applicable to QDC_SOLID.

9.2.2 Material Data

Material data must be defined for the element group. Consult Chapter 10 for the required input of the individual material models. Note that for some elements, it may be required that both a stress_material_model and a heat_conduction_model or a scalar_diffusion_model be prescribed.

9.2.3 Geometric Data

GEOMETRIC_MODEL

GEOMETRIC_MODEL file_name = "<string>" , etc...

Define the geometry for the continum elements. Two options are available. The data may be read in using keywords or as a list (optionally from another file).

Note	Variable Name	Type	Default	Description
	File_name	string	[none]	File name (optional). Name must be enclosed in quotation marks.
	Input_format **keywords** / list	list	[*]	Select input format option

9.2.3.1 Geometric / Material Properties

Note	Variable Name	Type	Default	Description
• *Keywords Read Method*				
	Geometric_set_number	integer	[1]	Geometric set number ≤ Number_of_geometric_sets
	Area	real	[0.0]	Cross section area
	Thickness	real	[0.0]	Thickness

• *List Read Method*

Geometric data must follow in the form:
< Geometric_set_number, Area, Thickness >.
< terminate with a blank record >.

9.2.4 Substrate Data□

SUBSTRATE_DATA

	SUBSTRATE_DATA	file_name = "<string>" , etc...
	Define the substrate data.	

Note	Variable Name	Type	Default	Description
	File_name	string	[none]	File name. Name must be enclosed in quotation marks.

9.2.4.1 Material Properties

Note	Variable Name	Type	Default	Description
	material_set_number	integer	[1]	Material set number ≤ number_of_material sets
(1)	material_type linear_elastic linear_viscous visco_elastic nonlinear_viscous generalized_visco_elastic elasto_plastic	list	[none]	Material type
	thickness_small_h	real	[0.0]	Thickness h
	thickness_big_H	real	[0.0	Thickness H
	viscosity	real	[0.0]	Viscosity η
	Maxwell_shear_modules	real	[0.0]	Shear modules μ_o
	Kelvin_shear_modules	real	[0.0]	Shear modules μ_∞
	number_of_coefficients	integer	[1]	Number of coefficients (≤ 2)
	mhu_1	real	[0.0]	Shear modules μ_1
	eta_1	real	[0.0]	Viscosity η_1

(cont'd)

(cont'd)

Note	Variable Name	Type	Default	Description
	mhu_2	real	[0.0]	Shear modules μ_2
	eta_2	real	[0.0]	Viscosity η_2
	power_exponent	real	[0.0]	Power exponent *n*
	Poissons_ratio	real	[0.0]	Poisson's ratio ν
	Yield_strength	real	[0.0]	Yield strength *Y*
	Thermal_expansion_coeff	real	[0.0]	Thermal expansion coefficient α
	Temperature_load_time	integer	[0]	Temperature load time function
	initial_shear_1	real	[0.0]	Initial shear stress τ_1^o
	initial_shear_2	real	[0.0]	Initial shear stress τ_2^o

Notes/
(1) The material may be selected as:

- Linear_elastic:

$$\tau_\alpha = \mu_o \, \gamma_\alpha$$

- Linear_viscous:

$$\tau_\alpha = \eta \frac{\partial \gamma_\alpha}{\partial t}$$

- Visco_elastic:

$$\begin{cases} \tau_\alpha = \mu_o \left(\gamma_\alpha - \gamma_\alpha^c \right) \\ \dfrac{\partial \gamma_\alpha^c}{\partial t} = \dfrac{\mu_o}{\eta} \left(\gamma_\alpha - \dfrac{\mu_o}{\mu_o - \mu_\infty} \gamma_\alpha^c \right) \end{cases}$$

- Nonlinear_viscous:

$$\tau_\alpha = \eta \frac{\partial \gamma_\alpha}{\partial t} \Big/ \dot{\gamma}_e^{\,1-\frac{1}{n}} \qquad \dot{\gamma}_e = \sqrt{\left(\frac{\partial \gamma_1}{\partial t}\right)^2 + \left(\frac{\partial \gamma_2}{\partial t}\right)^2}$$

- Generalized_visco_elastic:

$$\begin{cases} \tau_\alpha = \mu_o \, \gamma_\alpha - \sum\limits_m \mu_m \, \gamma_\alpha^m \\ \dfrac{\partial \gamma_\alpha^m}{\partial t} = \dfrac{\mu_m}{\eta_m}\left(\gamma_\alpha - \gamma_\alpha^m\right) \end{cases}$$

with

$$\mu_o = \mu_\infty + \sum_m \mu_m$$

- Elasto_plastic:

$$f = \sqrt{\sigma^2 + 3\tau_1^2 + 3\tau_2^2} - Y$$

with

$$\sigma^e = 2 \, \frac{\mu_o (1+\nu)}{1-\nu} \, \varepsilon \qquad \text{(elastic stress)}$$

$$\varepsilon = \Delta T \qquad \text{(thermal elastic strain)}$$

$$\tau_\alpha^e = \mu_o \gamma_\alpha \qquad \text{(elastic stress)}$$

9.2.5 Body Force Data (units: L/T^2)

BODY_FORCE

BODY_FORCE b_x1 = b(1) , ... etc

Note	Variable Name	Type	Default	Description
(1)	b_x1	real	[0.0]	Body force component in the x_1 direction
	b_x2	real	[0.0]	Body force component in the x_2 direction
	b_x3	real	[0.0]	Body force component in the x_3 direction
	b_force	real	[0.0]	Body force (scalar case)
	h_source	real	[0.0]	Heat source
(2)	injection_rate_phase_i	real	[0.0]	mass source/sink in phase i

Notes/

(2) Body force load multipliers are used to define the components of the gravity vector **b** with respect to the global (x_1, x_2, x_3) coordinate system, e.g., in SI units, **b** = {0.0, -9.81, 0.0} for the case x_2 vertical and oriented positively upward, with g = 9.81 m/s^2 and $\rho_w = 10^3$ kg/m^3.

(2) Only applicable to multi-phase flows.

9.2.6 Well Data

9.2.6.1 Injection Well Data

INJECTION_WELLS

INJECTION_WELLS file_name = "<string>", etc…
iwell, ne, ltime, icomp, iphase, fwell, etc...
< etc..., terminate with a blank record >.

Define the injection wells.

Note	Variable Name	Type	Default	Description
	File_name	string	[none]	File name. Name must be enclosed in quotation marks.
	Input_format keywords **list**	list	[*]	Input format option

Note	Variable Name	Type	Default	Description
	well_number	integer	[0]	Well number, (iwell ≤ Nwells)
	element_number	integer	[0]	Element number, (ne)
	node_number	integer	[0]	Node number (inode)
	load_time_function_number	integer	[0]	Load time function number, (ltime)
	component_name	string	[none]	Component name. Name must be enclosed in quotation marks
	injection_unit total_mass **mass_per_unit_volume**	list	[*]	Injection unit (iunit) iunit=1 iunit=0
	injection_rate	real	[0.0]	Injection rate, (fwell) fwell>0 : injection fwell<0 : extraction

- *List Input Case*
  Injection well data must follow in the form:
  < iwell, ne, ltime, inode, jcomp, iphase, iunit, fwell, inc, ne_last, iwell = 1, Nwells >
  < terminate with a blank record >

9.2.6.2 Production Well Data

PRODUCTION_WELLS

	PRODUCTION_WELLS iwell, ne, ltime, etc… < etc..., terminate with a blank record >.	file_name = "<string>", etc…

Define the production wells.

Note	Variable Name	Type	Default	Description
	File_name	string	[none]	File name. Name must be enclosed in quotation marks.
	Input_format keywords **list**	list	[*]	Input format option

Note	Variable Name	Type	Default	Description
	well_number	integer	[0]	Well number, (iwell ≤ Nwells)
	element_number	integer	[0]	Element number, (ne)
	load_time_function_number	integer	[0]	Load time function number, (ltime)
	bottom_hole_pressure	real	[0.0]	Bottom hole pressure (p_{wb})
	skin_factor	real	[0.0]	Skin factor (s_w)
	well_radius	real	[0.0]	Well radius (r_w)
	well_length	real	[0.0]	Well length (H_w)
	well_axis_e_1	real	[0.0]	Well axis component 1 (e_1)
	well_axis_e_2	real	[0.0]	Well axis component 2 (e_2)
	well_axis_e_3	real	[-1.0]	Well axis component 3 (e_3)

- *List Input Case*
  Injection well data must follow in the form:
  < iwell, ne, ltime, p_{wb}, s_w, r_w, H_w, e_1, e_2, e_3, inc, ne_last, iwell = 1, Nwells >
  < terminate with a blank record >

Notes/

Production wells may operate on deliverability against a prescribed flowing bottom hole pressure p_{wb} with a productivity index PI. With this option, the mass production rate of phase β from within an element with phase pressure p_β is:

$$q_\beta = - \frac{k_{r\beta}}{\mu_\beta} \rho_\beta \bullet PI \bullet \left(p_\beta - p_{wb} \right)$$

where

$$PI = 2\pi \frac{k H_w}{Ln \frac{r}{r_w} + s_w - \frac{1}{2}}$$

r_w = well radius, H_w = well length, s_w = skin factor, k = permeability (e.g., for a vertical well along the z-axis, $k = \sqrt{k_x k_y}$), and r = element radius (e.g., $r = \sqrt{h_x h_y / \pi}$).

The rate of production of component j is computed as:

$$q^j = \sum_\beta X^j_\beta \; q_\beta$$

where X^j_β = mass fraction of component j in phase β.

9.2.7 Fluid Cell Pressures Data

CELL_PRESSURE_LOADS

CELL_PRESSURE_LOADS
load_case_number = le, etc….

Define the fluid cell pressure load cases.

Note	Variable Name	Type	Default	Description
	load_case_number	integer	[0]	load case number
	load_time_function_number	integer	[0]	load time function number
	scaling_factor	real	[1.0]	scaling factor
	File_name	string	[none]	File name. Name must be enclosed in quotation marks.

- cell pressure data must follow in the form:
  <pressure (*i*), *i*=1, Numel>
  <terminate with a blank record>

9.2.8 Nodal Connectivity Data

Consult Chapter 11 for details. For this element NEN = number of nodes used to define the element (see Fig. 9.2.0.1). For instance: NEN = 2 in one-dimension, NEN = 4 in two-dimension, and NEN = 8 in three-dimension.

9.2.9 Output History Requests

FIELD_OUTPUT

FIELD_OUTPUT
n, ng, ntemp(1), ntemp(2), etc...
< etc..., terminate with a blank record >.

Plots of various element response components may be obtained. Each component requested is plotted versus time. Plots of this type are useful in providing quick information concerning the time history behavior of important data. The total number of components to be plotted must equal Number_output_sets, which is defined on the element group control command (see Section 9.1.1).

Note	Variable	Default	Description
(1)	N	[0]	Element number ≥ 1 and ≤ NUMEL
(2)	NG	[0]	Generation increment ≥ 0
(3)	NTEMP(1)	[0]	Component number ≥ 1 and ≤ NCOMP
	NTEMP(2)	[0]	Component number ≥ 1 and ≤ NCOMP
	etc.	. . .	. . .
	NTEMP(8)	[0]	Component number ≥ 1 and ≤ NCOMP

Notes/
(1) Element components history output data must be input for elements at which the time history of one or more components is to be plotted. *Terminate with a blank record.*

(2) Element components history output data can be generated by employing a two record sequence as follows:

Record 1: L, LG, LTEMP(1),..., LTEMP(8)
Record 2: N, NG, NTEMP(1),..., NTEMP(8)

The output time history requests of all elements:

L+LG, L+2*LG,..., N-MOD(N-L,LG)

(i.e., less than N) are set equal to those of element L. If LG is zero, no generation takes place between L and N.

(3) Up to three (1D case), seventeen (2D case) and nineteen (3D case) different component numbers may be plotted. The corresponding component numbers and output labels are as follows.

Notes from 9.2.9 (cont'd)

Table 9.2.9

One Dimensional Kinematics (NCOMP=3)

Component Number	Description	Output Label
1	Normal stress 11	S11
2	Strain 11	E11
3	Fluid pressure/Temperature	PF

Two Dimensional Kinematics (NCOMP=17)

Component Number	Description	Output Label
1	Normal stress 11	S11
2	Normal stress 22	S22
3	Normal stress 33	S33
4	Shear stress 12	S12
5	Principal stress 1	PS1
6	Principal stress 2	PS2
7	Shear stress	PTAU
8	Stress angle (between PS1 and X1)	SANG
9	Strain 11	E11
10	Strain 22	E22
11	Strain 33	E33
12	Engineering shear strain 12	G12
13	Principal strain 1	PE1
14	Principal strain 2	PE2
15	Engineering shear strain	PGAM
16	Strain angle (between PE1 and X1)	EANG
17	Fluid pressure/Temperature	PF

Notes from 9.2.9 (cont'd)

Three Dimensional Kinematics (NCOMP=19)

Component Number	Description	Output Label
1	Normal stress 11	S11
2	Normal stress 22	S22
3	Normal stress 33	S33
4	Shear stress 12	S12
5	Shear stress 23	S23
6	Shear stress 31	S31
7	Principal stress 1	PS1
8	Principal stress 2	PS2
9	Principal stress 3	PS3
10	Strain 11	E11
11	Strain 22	E22
12	Strain 33	E33
13	Engineering shear strain 12	G12
14	Engineering shear strain 23	G23
15	Engineering shear strain 31	G31
16	Principal strain 1	PE1
17	Principal strain 2	PE2
18	Principal strain 3	PE3
19	Fluid pressure/Temperature	PF

Notes . .

Notes . .

Notes . .

9.3 Structural Elements

9.3.0 Analysis Options

9.3.0.1 Truss Element

Truss elements connect two points in space and transmit axial forces only. There are three possible degrees of freedom at each of the two nodes, i.e. the x_1, x_2, and x_3 translations.

9.3.0.1.1 Linear Truss Element

LINEAR_TRUSS

Element_name = LINEAR_TRUSS , etc...
< stress model data >
< geometric data >
< body force data >
< connectivity data >
< field output data >

In the current implementation this truss element is restricted to linear isotropic elastic stress_models.

9.3.0.1.2 Linear / Nonlinear Truss Element

NONLINEAR_TRUSS

Element_name = NONLINEAR_TRUSS , etc...
< stress model data >
< geometric data >
< body force data >
< connectivity data >
< field output data >

The nonlinear truss element can be used with most elasto-plastic stress_models (except the pressure dependent material models). The nonlinear truss element can also be used with linear isotropic elastic stress_models.

9.3.0.2 Beam Element

Beam elements connect two points in space and transmit forces (axial and shear) and moments (torsion and bending). In 2D analysis there are three possible degrees of freedom at each of the two nodes, i.e., two translations and one rotation. In 3D analysis there are six possible degrees of freedom at each of the two nodes, i.e., three translations and three rotations. The local sign convention for the beam element is shown in Fig. 9.3.0.2.1.

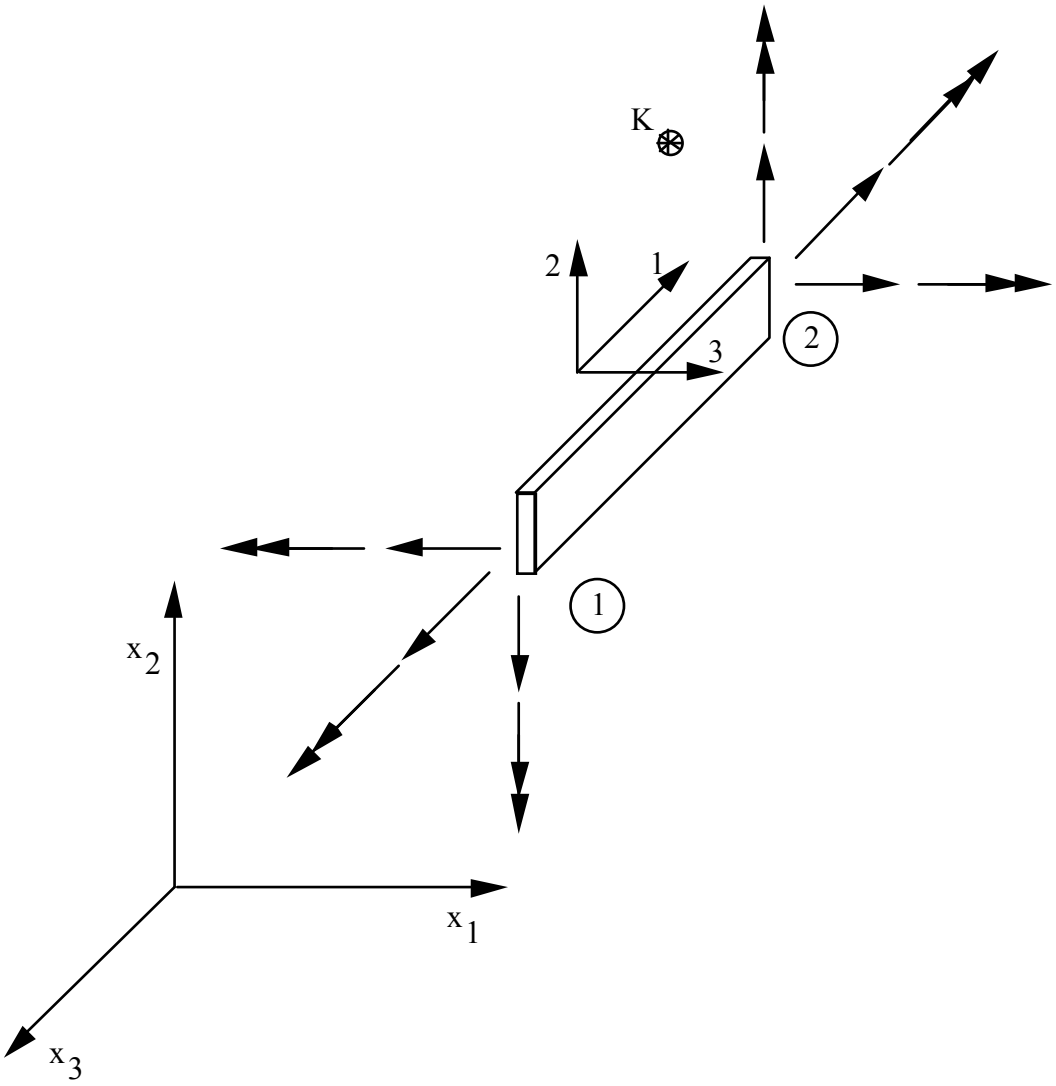


Figure 9.3.0.2.1 Local Coordinate System for Beam Element

9.3.0.2.1 Linear Beam Element

LINEAR_BEAM

Element_name = LINEAR_BEAM , etc...
< stress model data >
< geometric data >
< body force data >
< connectivity data >
< field output data >

In the current implementation the linear beam element is restricted to linear isotropic elastic stress_models.

9.3.0.2.2 Linear / Nonlinear Beam Element

NONLINEAR_BEAM

Element_name = NONLINEAR_BEAM , etc...
< stress model data >
< geometric data >
< body force data >
< connectivity data >
< field output data >

The nonlinear beam element can be used with most elasto-plastic stress_models (except the pressure dependent material models). The nonlinear beam element can also be used with linear isotropic elastic stress_models.

9.3.0.3 Plate and Shell Elements

The elements must be used in quadrilateral form for 2D plate (flat) and 3D plate/shell analysis. The local sign convention for the plate/shell elements is shown in Fig. 9.3.0.3.1.

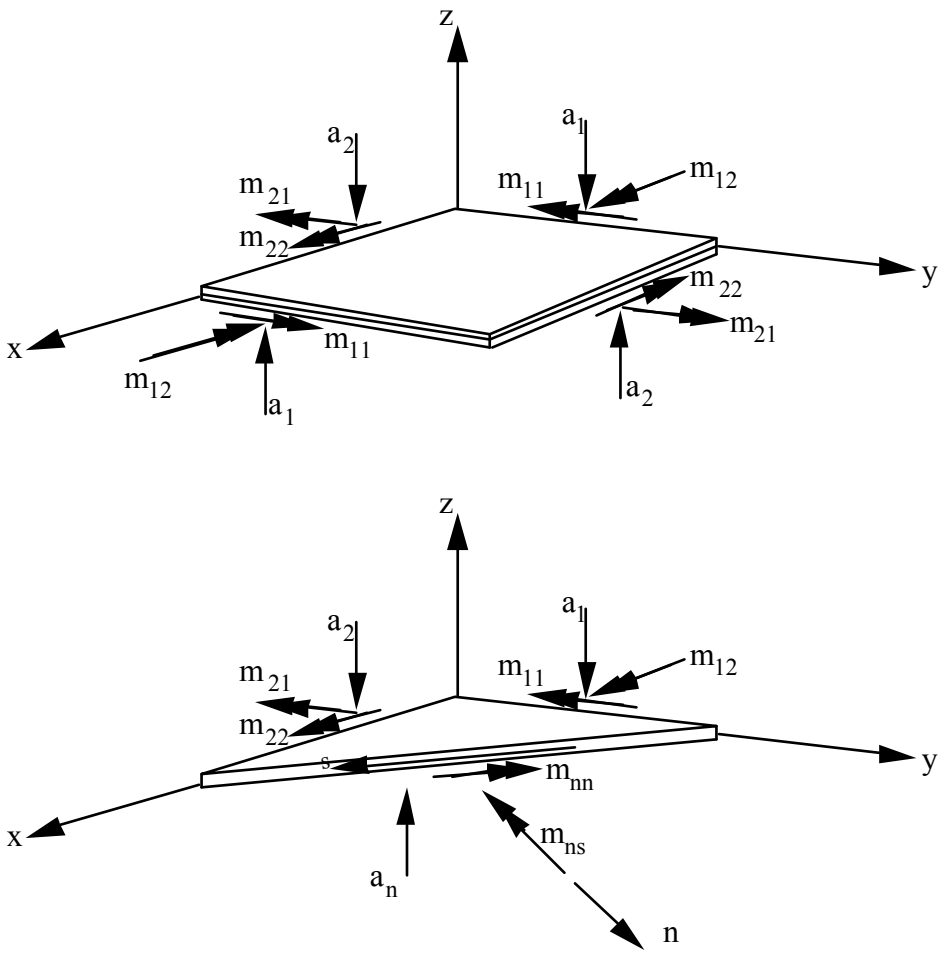


Figure 9.3.0.3.1 Sign Convention for Stress Resultants for Plate/Shell Elements

9.3.0.3.1 Plate / Shell Element

PLATE

Element_name = PLATE , etc...
< stress model data >
< geometric data >
< body force data >
< connectivity data >

The element must be used in quadrilateral form for 2D plate (flat) and 3D plate/shell analysis. In 2D analysis there are three possible degree of freedom at each nodes, i.e., one vertical translation and two rotations. In 3D analysis there are six possible degrees of freedom at each nodes, i.e., three translations and three rotations. The local sign conven- tion for the plate/shell element is shown in Fig. 9.3.0.3.1. In the current implementation the plate element is restricted to linear isotropic elastic stress_models.

9.3.0.3.2 Shell / Plate Element

SHELL_PLATE

Element_name = SHELL_PLATE , etc...
< stress model data >
< geometric data >
< body force data >
< connectivity data >

The element must be used in quadrilateral form for 3D shell/plate analysis. There are six possible degrees of freedom at each nodes, i.e., three translations and three rotations. The local sign convention for the shell/plate element is shown in Fig. 9.3.0.3.1. In the current implementation the shell_plate element is restricted to linear isotropic elastic stress_models.

9.3.0.3.3 Bilinear Shell Element

SHELL_BILINEAR

Element_name = SHELL_BILINEAR , etc...
< stress model data >
< geometric data >
< body force data >
< connectivity data >

The element must be used in quadrilateral form for 3D shell analysis. There are five possible degrees of freedom at each nodes, i.e., three translations and two rotations. The local sign convention for the shell element is shown in Fig. 9.3.0.3.1. This shell element is based on the formulation presented by Ref [1]. The implementation was performed by Swan Colby in AY 91-92. The coordinates of the nodes on the midsurfaces of the shell must be input. The nodal degrees of freedom 4 and 5 are *local* degrees of freedom, and correspond to the rotations of the fiber about the local basis vectors, (see Fig. 6.2.5 pp.388 in Ref [1]). In the current implementation the shell_plate element is restricted to linear isotropic elastic stress_models.

References / Bibliography

1. Hughes, T.J.R., *The Finite Element Method*, Prentice Hall, Englewood Cliffs. NJ. (1987).

9.3.0.4 Membrane Element

MEMBRANE

Element_name = MEMBRANE , etc...
< stress model data >
< geometric data >
< body force data >
< connectivity data >

The element must be used in quadrilateral form. There are three possible degrees of freedom at each node, i.e., three translations. In the current implementation the membrane element is restricted to isotropic elastic stress_models.

9.3.1 Element Control Information

Note	Variable Name	Type	Default	Description
	Element_name	list	[none]	Element name
	Element_shape	list	[none]	Element shape
	• *Selection for Plate, Shell or Membrane Elements* Four_node_quad			
	• *Selection for Truss or Beam Elements* Two_node_line			
	Finite_deformation on / **off**	list	[off]	Finite deformation option
	Geometric_stiffness on / **off**	list	[off]	Geometric stiffness option
	Numerical_integration **full** / reduced	list	[full]	Numerical integration option
	Bending_integration f**ull** / reduced	list	[full]	Bending integration option
	Shear_integration full / **reduced**	list	[reduced]	Shear integration option
	Membrane_integration standard **bbar** reduced	list	[bbar]	Membrane integration option: Standard formulation Selective-reduced integration Reduced integration
	Fiber_integration	integer	[2]	Number of fiber integration point: ≥ 1 and ≤ 5
	Mass_type **lumped** / consistent	list	[lumped]	Mass type
	Number_of_geometric_sets	integer	[1]	Number of geometric sets ≥ 1
	Body_force_load_time	integer	[0]	Body force load-time function number

EXAMPLE

```
Element_group   /
        name = "group 1"   /
        element_type = structural  /
        element_shape = two_node_line  /
        element_name = linear_beam  /
        number_of_output_sets = 4  /
        number_of_material_sets = 1  /
        number_of_geometric_sets = 1
```

EXAMPLE (cont'd)

```
Stress_model  /
    material_name = linear_elastic  /
    material_type = linear

        material_set_number = 1  /
        mass_density = 1.E-2  /
        youngs_modulus = 100.  /
        poissons_ratio = 0.0

Geometric_model
        geometric_set_number = 1  /
        area = 5.0  /
        bending_inertia = 10.

Nodal_connectivity
          1      1      1      2      1
         20      1      1

Field_output
          1      0      1      4
         10      0      1      4
```

9.3.2 Material Data

Material data must be defined for the element group. Consult Chapter 10 for the required input of the individual stress models. Note that not all material models are applicable to the structural elements.

9.3.3 Geometric Data

GEOMETRIC_MODEL

GEOMETRIC_MODEL file_name = "<string>" , etc...

Define the geometry for the structural elements. Two options are available. The data may be read in using keywords or as a list (optionally from another file).

Note	Variable Name	Type	Default	Description
	File_name	string	[none]	File name (optional). Name must be enclosed in quotation marks.
	Input_format **keywords** / list	list	[*]	Select input format option

9.3.3.1 Geometric / Material Properties

Note	Variable Name	Type	Default	Description
• *Keywords Read Method*				
	Geometric_set_number	integer	[1]	Geometric set number ≤ Number_of_geometric_sets
• *Truss Elements*				
	Area	real	[0.0]	Cross section area
• *Beam Elements*				
	Area	real	[0.0]	Cross section area
	Shear_area_2	real	[0.0]	Effective shear area, direction 2
	Shear_area_3	real	[0.0]	Effective shear area, direction 3
	Inertia_I11	real	[0.0]	Torsional moment of inertia I_{11}
	Inertia_I22	real	[0.0]	Transverse moment of inertia I_{22}
	Inertia_I33	real	[0.0]	Bending moment of inertia I_{33}
	Height	real	[0.0]	Beam height
	Width	real	[0.0]	Beam width
	Web_thickness	real	[0.0]	Web thickness
	Flange_thickness	real	[0.0]	Flange thickness
	Cross_section_type	string	[*]	Cross section type
	I_beam			I Beam
	Hollow_box			Hollow box
	Rectangular			Rectangular
	Ref_coord_x1	real	[0.0]	Reference point K, coordinate x_1
	Ref_coord_x2	real	[0.0]	Reference point K, coordinate x_2
	Ref_coord_x3	real	[0.0]	Reference point K, coordinate x_3

(cont'd)

(cont'd)

Note	Variable Name	Type	Default	Description
• *Plate, Shell and Membrane Elements*				
	Thickness	real	[0.0]	Thickness
• *List Read Method*				
	Geometric data must follow in the form: < Geometric_set_number, Area or Thickness, Inertia_I11, Inertia_I22, Inertia_I33, Ref_coord_x1, Ref_coord_x2, Ref_coord_x3, Shear_area_2, Shear_area_3 > < terminate with a blank record >.			

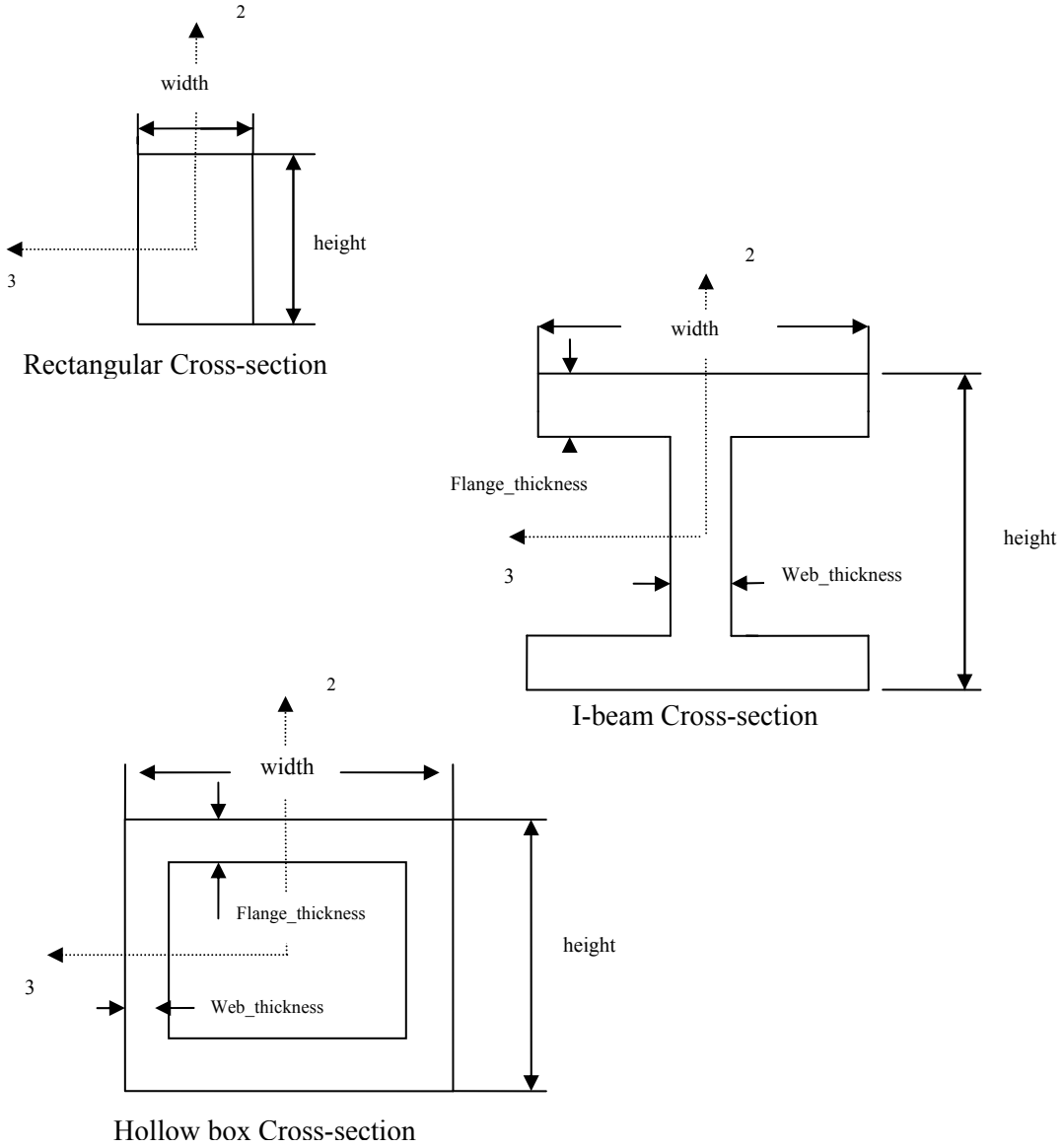


Figure 9.3.3.1.1 Beam Cross-Sections

9.3.4 Prestressing

PRESTRESSING

	PRESTRESSING	file_name = "<string>" , etc...	

Define prestressing forces in beam elements. Two options are available. The data may be read in using keywords, or as a list (optionally from another file).

Note	Variable Name	Type	Default	Description
	File_name	string	[none]	File name (optional). Name must be enclosed in quotation marks.
	Input_format keyword / **list**	list	[*]	Select input format option.

9.3.4.1 Prestressing Forces

Note	Variable Name	Type	Default	Description
• *Keywords Read Method*				
	Element_number	integer	[0]	Element number
	Prestress_force	real	[0.0]	Prestressing force (tension positive)
(1)	Eccentricity	real	[0.0]	Eccentricity

• *List Read Method*

Prestressing data must follow in the form:

< element_number, prestress_force (element_number), eccentricity (element_number) >

< terminate with a blank record >.

Note/
(1) Assumes a tendon with a straight profile.

9.3.5 Pretension

PRETENSION

	PRETENSION	file_name = "<string>" , etc...

Define pretension forces in beam and truss elements. Two options are available. The data may be read in using keywords, or as a list (optionally from another file).

Note	Variable Name	Type	Default	Description
	File_name	string	[none]	File name (optional). Name must be enclosed in quotation marks.
	Input_format keyword / **list**	list	[*]	Select input format option.

9.3.5.1 Pretension Forces

Note	Variable Name	Type	Default	Description
• *Keywords Read Method*				
	Element_number	integer	[0]	Element number
	Pretension_force	real	[0.0]	Pretension force (tension positive)
• *List Read Method*				
	Pretension data must follow in the form: < element_number, pretension_force (element_number) > < terminate with a blank record >.			

9.3.6 Body Force Data (units: L/T^2)

BODY_FORCE

	BODY_FORCE	b_x1 = b(1) , ..etc...

Note	Variable Name	Type	Default	Description
(1)	b_x1	real	[0.0]	Body force component in the x_1 direction
	b_x2	real	[0.0]	Body force component in the x_2 direction
	b_x3	real	[0.0]	Body force component in the x_3 direction

Notes/
(1) Body force load multipliers are used to define the components of the gravity vector **b** with respect to the global (x_1, x_2, x_3) coordinate system, e.g., in SI units, **b** = {0.0, -9.81, 0.0} for the case x_2 vertical and oriented positively upward, with g = 9.81 m/s^2 and $\rho_w = 10^3$ kg/m^3.

9.3.7 Nodal Connectivity Data

Consult Chapter 11 for details. For this element NEN = number of nodes used to define the element; viz. NEN = 2 for Truss and Beam elements, NEN = 4 for Plate, Shell and Membrane elements.

9.3.8 Output History Requests

FIELD_OUTPUT

FIELD_OUTPUT
n , ng , ntemp(1) , ntemp(2) , etc…
< etc..., terminate with a blank record >

Plots of various element response components may be obtained. Each component requested is plotted versus time. Plots of this type are useful in providing quick information concerning the time history behavior of important data. The total number of components to be plotted must equal Number_output_sets, which is defined on the element group control command (see Section 9.3.1).

Note	Variable	Default	Description
(1)	N	[0]	Element number ≥ 1 and ≤ NUMEL
(2)	NG	[0]	Generation increment ≥ 0
(3)	NTEMP(1)	[0]	Component number ≥ 1 and ≤ NCOMP
	NTEMP(2)	[0]	Component number ≥ 1 and ≤ NCOMP
	.	.	.
	etc.	.	.
	.	.	.
	NTEMP(8)	[0]	Component number ≥ 1 and ≤ NCOMP

Notes/
(1) Element components history output data must be input for elements at which the time history of one or more components is to be plotted. *Terminate with a blank record.*

(2) Element components history output data can be generated by employing a two record sequence as follows:

Record 1: L, LG, LTEMP(1),..., LTEMP(8)
Record 2: N, NG, NTEMP(1),..., NTEMP(8)

The output time history requests of all elements:

L+LG, L+2*LG,..., N-MOD(N-L,LG)

(i.e., less than N) are set equal to those of element L. If LG is zero, no generation takes place between L and N.

(3) The corresponding component numbers and output labels are as follows:

Notes from 9.3.8 (cont'd)

Table 9.3.8.1

Truss (NCOMP=2)

Component Number	Description	Output Label
1	Axial Stress 11	STRS
2	Axial Force	FORC

Two Dimensional Beam (NCOMP=6)

Component Number	Description	Output Label
1	Axial force, Node 1	N1-1
2	Shear force, Node 1	N2-1
3	Bending Moment, Node 1	M3-1
4	Axial force, Node 2	N1-2
5	Shear force, Node 2	N2-2
6	Bending Moment, Node 2	M3-2

Three Dimensional Beam (NCOMP=12)

Component Number	Description	Output Label
1	Axial force, Node 1	N1-1
2	Shear force 2, Node 1	N2-1
3	Shear force 3, Node 1	N3-1
4	Torsion Moment, Node 1	M1-1
5	Bending Moment 2, Node 1	M2-1
6	Bending Moment 3, Node 1	M3-1
7	Axial force, Node 2	N1-2
8	Shear force 2, Node 2	N2-2
9	Shear force 3, Node 2	N3-2
10	Torsion Moment, Node 2	M1-2
11	Bending Moment 2, Node 2	M2-2
12	Bending Moment 3, Node 2	M3-2

Notes from 9.3.8 (cont'd)

Table 9.3.8.2

Plate and Shell (NCOMP=8)

Component Number	Description	Output Label
1	Bending Moment, m_{11}	M12
2	Bending Moment, m_{22}	M22
3	Bending Moment, m_{12}	M12
4	Shear Force, q_1	Q1
5	Shear Force, q_1	Q2
6	Membrane Stress, σ_{11}	S11
7	Membrane Stress, σ_{22}	S22
8	Membrane Stress, σ_{12}	S12

Membrane (NCOMP=6)

Component Number	Description	Output Label
1	Normal Stress, σ_{11}	S11
2	Normal Stress, σ_{22}	S22
3	Shear Stress, σ_{12}	S12
4	Normal Strain, ε_{11}	E11
5	Normal Strain, ε_{22}	E22
6	Shear Strain, γ_{12}	G12

Notes . .

9.4 Interface Surface Element

INTERFACE_SURFACE

Element_name = INTERFACE_SURFACE, etc.
m, phi(m), c(m) < m = 1, numat >
< slaved nodes data >
< master elements connectivity data >
< exterior surfaces connectivity data >

9.4.1 Element Group Control Information

Must follow the element name (same data record), and define the control parameters in the following form:

Note	Variable Name	Type	Default	Description
	Number_of_slave_nodes	integer	[0]	Number of slave nodes
	Number_of_exterior_surfaces	integer	[0]	Number of exterior surfaces
	Analysis_type **two_dimensional** three_dimensional	list	[*]	Analysis type

9.4.2 Geometric / Material Properties Data (Numat sets)

Note	Variable	Default	Description
	M	[0]	Geometric/material set number
	PHI(M)	[0.0]	Friction angle (degrees)
	C(M)	[0.0]	Cohesion

9.4.3 Slaved Nodes Data

Note	Variable	Default	Description
	N	[0]	Node number ≥ 1 and ≤ NUMNP
(1)	NG	[0]	Generation increment

Notes/

(1) Slaved nodal data can be generated by employing a two record sequence as follows:

Record 1: L,LG
Record 2: N,NG,

The nodes:
L+LG, L+2*LG,...,N-MOD(N-L,LG)

(i.e., less than N) are then defined as slaved. If LG is blank or zero, no generation takes place between L and N.

9.4.4 Master Elements Nodal Connectivity Data

Consult Chapter 11 for details; for this element NEN = 8 in three-dimensions, and NEN = 4 in two-dimensions.

9.4.5 Exterior Surfaces Nodal Connectivity Data

Consult Chapter 11 for details; for this case NEN = 4 in three-dimensions, and NEN = 2 in two-dimensions.

References / Bibliography

1. Belytschko, T. and Lin, J.I., "A three-dimensional impact-penetration algorithm with erosion", *Computers and Structures*, **32**, (1987), 95-104.

2. Hallquist, J.O., "A procedure for the solution of finite-deformation contact-impact problems by the finite element method", UCRL-52066, Lawrence Livermore Laboratory, Livermore, CA, (1976).

3. Hallquist, J.O., "A numerical procedure for three dimensional impact problems", Preprint 2956, ASCE, (October 1977).

4. Hallquist, J.O., "A numerical treatment of sliding interfaces and impact", *Computational Techniques for Interface Problems*, Eds K.C. Park and D.K. Gartling, ASME, NY, **AMD 30**, (1978).

Notes . .

Notes . .

9.5 Contact Surface Element

The contact element may be used to impose inequality constraints between nodes. Either perfect friction (i.e. "stick") or frictionless (i.e. "slip") conditions may be achieved. A contact element is defined by two nodes a spring constant, or "penalty parameter," k; and a fixed direction vector, N. The present location of node A (A = 1, 2) is given by X(A) + D(A) where X(A) is the initial position vector and D(A) is the displacement vector. The contact plane passes through the point X(A) + D(A) and is perpendicular to N (see Fig. 9.5.1).

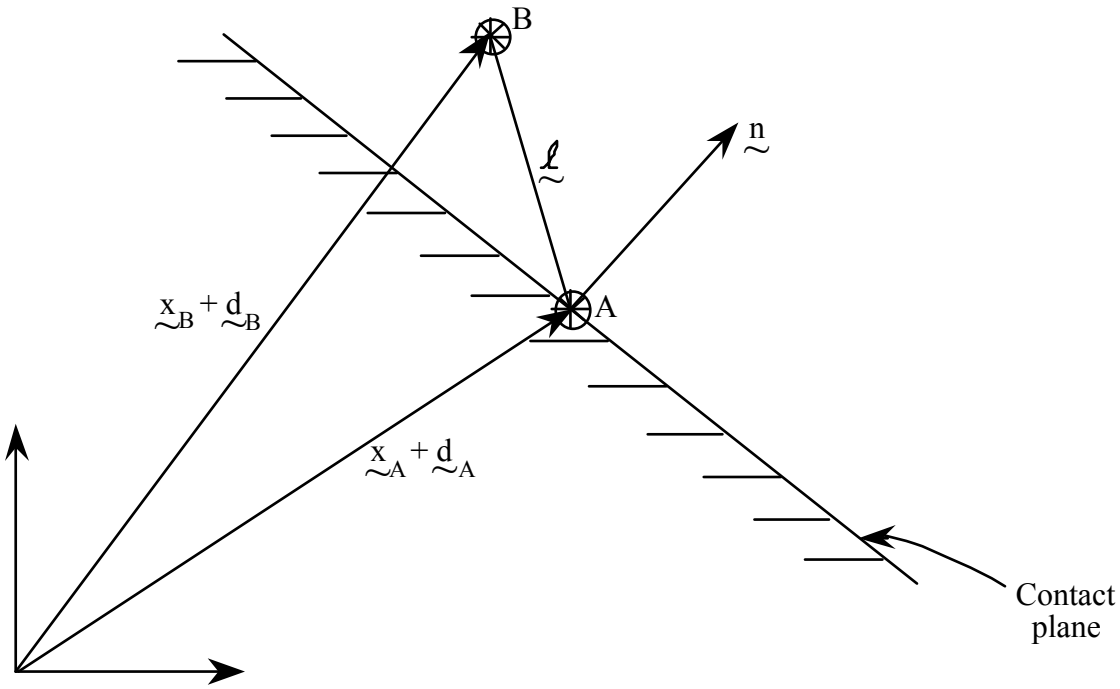


Figure 9.5.1 Contact Surface

The contact/release condition is defined as follows

$$d > 0 \quad \text{release} \qquad d \le 0 \quad \text{contact}$$

where

$$d = \mathbf{L} \cdot \mathbf{N}$$

$$\mathbf{L} = (\mathbf{X}(B) + \mathbf{D}(B)) - (\mathbf{X}(A) + \mathbf{D}(A))$$

The quantity d is a measure of the distance between X(B) + D(B) and the contact plane. When contact is noted, a contact element stiffness and out-of-balance force are added to the global equations. These arrays are defined as follows:

Stiffness Matrices

(two-dimensional case)

$$K(\text{stick}) = k \begin{bmatrix} 1 & 0 & -1 & 0 \\ & 1 & 0 & -1 \\ & & 1 & 0 \\ \text{Symm.} & & & 1 \end{bmatrix}$$

$$K(\text{slip}) = k \begin{bmatrix} n1n1 & n1n2 & -n1n1 & -n1n2 \\ & n2n2 & -n1n2 & -n2n2 \\ & & n1n1 & n1n2 \\ \text{Symm.} & & & n2n2 \end{bmatrix} = k \begin{bmatrix} -N \\ N \end{bmatrix} \begin{bmatrix} -N \\ N \end{bmatrix}^T$$

(three-dimensional case)

$$K(\text{stick}) = k \begin{bmatrix} 1 & 0 & 0 & -1 & 0 & 0 \\ & 1 & 0 & 0 & -1 & 0 \\ & & 1 & 0 & 0 & -1 \\ & & & 1 & 0 & 0 \\ & & & & 1 & 0 \\ \text{Symm.} & & & & & 1 \end{bmatrix}$$

$$K(slip) = k \begin{bmatrix} n1n1 & n1n2 & n1n3 & -n1n1 & -n1n2 & -n1n3 \\ & n2n2 & n2n3 & -n1n2 & -n2n2 & -n2n3 \\ & & n3n3 & -n1n3 & -n2n3 & -n3n3 \\ & & & n1n1 & n1n2 & n1n3 \\ & Symm. & & & n2n2 & n2n3 \\ & & & & & n3n3 \end{bmatrix}$$

Out-of-Balance Force

$$f = -K(slip) \begin{bmatrix} X(A) + D(A) \\ X(B) + D(B) \end{bmatrix} = kD \begin{bmatrix} N \\ -N \end{bmatrix}$$

If $k > 0$ is sufficiently large, the point X(B) + D(B) will be forced to lie (approximately) on the contact plane. In subsequent steps, only the contact stiffness is assembled and the decision to remain in contact, or release, is made on the basis of the sign of d, as above.

For purposes of interpreting output, the contact element "displacement" is defined to be d, and the "force" is given by

$k.d$ if $d < 0$

0 if $d \geq 0$

CONTACT_SURFACE

Element_name = CONTACT_SURFACE stick = on, etc...
m, stiff(m), (an(i, m), i = 1, nsd) < m = 1, numat >
< connectivity data >
< field output data >
< terminate with a blank record >.

9.5.1 Element Group Control Information

Must follow the element name (same data record), and define the control parameters as follows:

Note	Variable Name	Type	Default	Description
(1)	stick **on** / off	list	[on]	Contact condition
	formulation **penalty** augmented_lagrangian	list	[penalty]	Formulation

Notes/
(1) The contact condition may be either perfect friction (i.e. "stick") or frictionless (i.e. "slip"). In the stick case, no relative motion is allowed between nodes when in contact. In the slip case, frictionless sliding is allowed in the contact plane.

9.5.2 Geometric / Material Properties Data (Numat sets)

Note	Variable	Default	Description
	M	[0]	Geometric/material set number
	STIFF(M)	[0.0]	Spring constant k (i.e. "penalty parameter")
	AN(1,M)	[0.0]	Component 1 of direction vector N
	AN(2,M)	[0.0]	Component 2 of direction vector N
	AN(3,M)	[0.0]	Component 3 of direction vector N

9.5.3 Element Nodal Connectivity Data

Consult Chapter 11 for details; for this element NEN = 2.

9.5.4 Element Output History Requests

"On-line" and Calcomp plots of various element response components may be obtained. Each component required is plotted versus time. Plots of this type are useful in providing quick information concerning the time history behavior of important data. The total number of components to be plotted must equal NOUT, which is defined on the element group control record (see Section 9.1.1).

Note	Variable	Default	Description
(1)	N	[0]	Element number ≥ 1 and ≤ Numel
(2)	NG	[0]	Generation increment ≥ 0
(3)	NTEMP(1)	[0]	Component number ≥ 1 and ≤ 2
	NTEMP(2)	[0]	Component number ≥ 1 and ≤ 2

Notes/
(1) Element components history output data must be input for elements at which the time history of one or more components is to be plotted. Records need not be read in order. *Terminate with a blank record.*

(2) Element components history output data can be generated by employing a two record sequence as follows:

Record 1: L,LG,LTEMP(1),LTEMP(2)
Record 2: N,NG,NTEMP(1),NTEMP(2)

The output time history requests of all elements

L+LG, L+2*LG,...,N-MOD(N-L,LG)

(i.e., less than N) are set equal to those of node L. If LG is blank or zero, no generation takes place between L and N.

(3) Output history information is stored in the array IHS in element group data. The dimension of IHS is 2 x NOUT. The first row of IHS contains element numbers and the second row contains output history component numbers. Two different component numbers may be plotted as described above. The corresponding component numbers and output labels are:

Component Number	Description	Output Label
1	Displacement	DELT
2	Contact force	FORC

Notes . .

9.6 Slide-Line Element

The slide-line element may be used to impose kinematic constraints between nodes. Either perfect friction (i.e., "stick") or frictionless (i.e., "slip") conditions may be achieved.

A slide-line element is defined by three nodes and a spring constant or "penalty parameter," k. The connection from node A to node B defines the "slide-line" direction, and node C is the contact node (see Fig. 9.6.1).

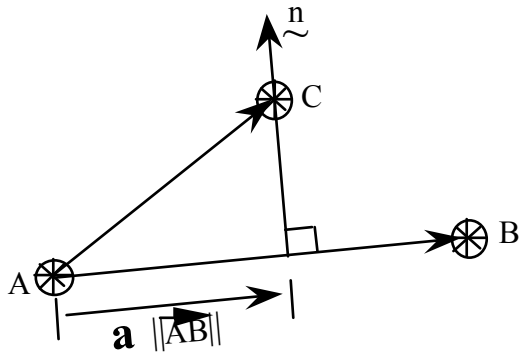


Figure 9.6.1 Slide Line

The projected distance of node C to node A onto the slide-line direction is denoted by a, and is given by

$$a = \mathbf{AB} \cdot \mathbf{AC} / (\mathbf{AB} \cdot \mathbf{AB}) \qquad 0 \le a \le 1$$

where "." denotes the dot product of two vectors. The direction of the unit vector n normal to the slide-line direction is given by:

$$\mathbf{n} = (\mathbf{AB} \times \mathbf{AC}) \times \mathbf{AB} / |(\mathbf{AB} \times \mathbf{AC}) \times \mathbf{AB}|$$

where "x" denotes the cross product of two vectors. The local contact stiffness matrix **K** is given by

$$K = k \begin{bmatrix} (1-a)(1-a) & a(1-a) & -(1-a) \\ & aa & -a \\ \text{Symm.} & & 1 \end{bmatrix}$$

where the rows and columns are arranged such that the first, second and third rows (columns) correspond to nodes A, B and C, respectively. If $k > 0$ is sufficiently large, the point C will be forced to lie (approximately) on the slide-line AB. The contact/release condition is defined as follows:

$0 \le a \le 1$ contact

otherwise, release.

When contact is noted, a contact element stiffness and out-of-balance force are added to the global equations. These arrays are defined as follows:

Stiffness Matrix: (three-dimensional case)

$$K(stick) = \begin{bmatrix} (1-a)(1-a) & 0 & 0 & a(1-a) & 0 & 0 & -(1-a) & 0 & 0 \\ & (1-a)(1-a) & 0 & 0 & a(1-a) & 0 & 0 & -(1-a) & 0 \\ & & (1-a)(1-a) & 0 & 0 & a(1-a) & 0 & 0 & -(1-a) \\ & & & aa & 0 & 0 & -a & 0 & 0 \\ & & & & aa & 0 & 0 & -a & 0 \\ & \text{Symm.} & & & & aa & 0 & 0 & -a \\ & & & & & & 1 & 0 & 0 \\ & & & & & & & 1 & 0 \\ & & & & & & & & 1 \end{bmatrix}$$

$K(slip) = R^T K R$

where

$$R = \begin{bmatrix} n^T & 0 & 0 \\ 0 & n^T & 0 \\ 0 & 0 & n^T \end{bmatrix} \qquad \text{and} \qquad 0 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$$

Out-of-Balance Force

$$f = -K(slip) \begin{bmatrix} X(A) + D(A) \\ X(B) + D(B) \\ X(C) + D(C) \end{bmatrix}$$

SLIDE_LINE

Element_name = SLIDE_LINE stick = on, etc...
m, stiff(m) < m = 1, numat >
  < connectivity data >
< terminate with a blank record >

9.6.1 Element Group Control Information

Must follow the element name (same data record), and define the control parameters as follows:

Note	Variable Name	Type	Default	Description
(1)	stick **on** / off	list	[on]	Contact condition
	formulation **penalty** augmented_lagrangian	list	[penalty]	Formulation

Notes/
(1) The contact condition may be either perfect friction (i.e. "stick") or frictionless (i.e. "slip"). In the stick case, no relative motion is allowed between nodes when in contact. In the slip case, frictionless sliding is allowed on the slide-line.

9.6.2 Geometric / Material Properties Data (Numat sets)

Note	Variable	Default	Description
	M	[0]	Geometric/material set number
	STIFF(M)	[0.0]	Spring constant k (i.e. "penalty parameter")

9.6.3 Element Nodal Connectivity Data

Consult Chapter 11 for details; for this element NEN = 3, and the nodes are entered in the following order; nodeA, nodeB, nodeC (see Figure 9.6.1).

Notes . .

9.7 Slide-Line Element with Coulomb Friction

The slide-line element is defined by three nodes and two spring constants or "penalty parameters," k1 and k2, in the tangential and normal directions, respectively. The connection from node A to node B defines the "slide-line" direction, and node C is the contact node (see Fig. 9.7.1).

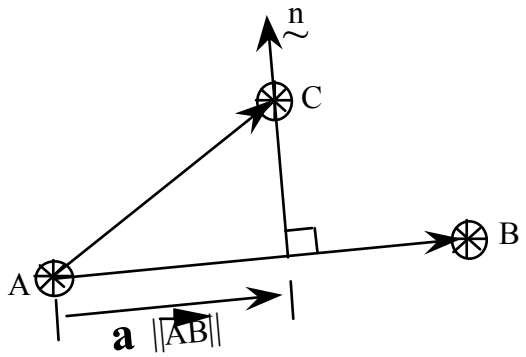


Figure 9.7.1 Slide Line with Coulomb Friction

The tangent vector is defined as:

$$\mathbf{t} = \mathbf{AB} / |\mathbf{AB}|$$

The direction of the unit vector **n** normal to the slide-line direction is given by:

in 2D: by rotating the tangent vector 90 degrees conterclockwise: $\mathbf{n} = \mathbf{e}_3 \times \mathbf{t}$

in 3D: $\mathbf{n} = -(\mathbf{AB} \times \mathbf{AC}) \times \mathbf{t} / |\mathbf{AB} \times \mathbf{AC}|$

where "**x**" denotes the cross product of two vectors. The projected distance of node C to node A onto the slide-line direction is denoted by a, and is given by

$$a = \mathbf{AB} \cdot \mathbf{AC} / |\mathbf{AB}|^2 = \mathbf{AC} \cdot \mathbf{t} / |\mathbf{AB}| \qquad 0 \le a \le 1$$

where "." denotes the dot product of two vectors. The relative normal displacement, or gap, is computed as:

$$g_n = |\mathbf{AB} \times \mathbf{AC}| / |\mathbf{AB}|^2 = \mathbf{AC} \cdot \mathbf{n} / |\mathbf{AB}|$$

and the relative slip as:

$$g_t = a - a_0$$

where a_0 is the relative position at which node C first contacted the line AB. The normal and tangential stresses are computed as:

$$S_n = k2 * g_n \quad \text{and} \quad S_t = k1 * g_t$$

The normal stress must be compressive, i.e.,

$$S_n \leq 0$$

and the tangential stress such that

$$|S_t| \leq \tan(\phi)*|S_n| \qquad (*)$$

where □= friction angle. The Coulomb friction law is associated with a no-slip condition and a directional constraint that requires the friction force to always act opposite to the direction of the relative slip of node C with respect to nodes A and B. A return procedure is used to enforce inequality (*) when violated.

The local contact stiffness matrix **K** is given by:

$$K = k \begin{bmatrix} (1\text{-}a)(1\text{-}a) & a(1\text{-}a) & -(1\text{-}a) \\ & aa & -a \\ \text{Symm.} & & 1 \end{bmatrix}$$

where k = k1 and k2 for the tangential and normal directions, respectively, and where the rows and columns are arranged such that the first, second and third rows (columns) correspond to nodes A, B and C, respectively. The contact/release condition is defined as follows:

$$\text{If } (0 \leq a \leq 1 \text{ and } S_n \leq 0) \text{ ===> contact}$$

otherwise, release.

When contact is noted, a contact element stiffness and out-of-balance force are added to the global equations, by rotating the local stiffness and force to the global axes.

SLIDE_COULOMB

Element_name = SLIDE_COULOMB
m, stiff(1, m), stiff(2, m), phi(m), c(m) < m = 1, numat >
< connectivity data >
< terminate with a blank record >.

9.7.1 Element Group Control Information

Must follow the element name (same data record), and define the control parameter as follows:

Note	Variable Name	Type	Default	Description
(1)	Gapping **on** / off	list	[on]	Gapping code (only applicable to 2D cases)
	Friction_load_time	integer	[0]	Friction angle load time function number
	formulation **penalty** augmented_lagrangian	list	[penalty]	Formulation

Notes /
(1) This allows the contact-release option to be deactivated if needed.

9.7.2 Geometric / Material Properties Data (Numat sets)

Note	Variable	Default	Description
	M	[0]	Geometric/material set number
	STIFF(1,M)	[0.0]	Spring constant k1
	STIFF(2,M)	[0.0]	Spring constant k2
	PHI(M)	[0.0]	Friction angle (degrees)
	C(M)	[0.0]	Cohesion

9.7.3 Element Nodal Connectivity Data

Consult Chapter 11 for details; for this element NEN = 3, and the nodes are entered in the following order; node A, node B, node C (see Figure 9.7.1).

Notes . .

Notes . .

Notes . .

9.8 Contact Plane with Coulomb Friction

The element is defined by four nodes and two spring constants or "penalty parameters", k_t and k_n in the tangential and normal directions, respectively. The nodes A, B and C define the contact plane, and node D is the contact node (see Figure 9.8.1).

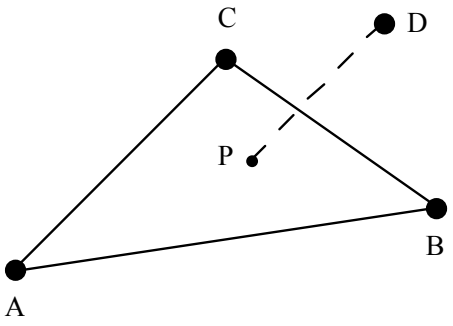


Figure 9.8.1 Contact Plane with Coulomb Friction

The direction of the unit vector $\boldsymbol{n}$ normal to the contact plane is defined as:

$$\boldsymbol{n} = \boldsymbol{AB} \times \boldsymbol{AC} / |\boldsymbol{AB} \times \boldsymbol{AC}|$$

where "$\times$" denotes the cross product of two vectors. The relative normal displacement, or gap, is computed as:

$$g_n = \boldsymbol{AD} \cdot \boldsymbol{n}$$

where "." denotes the dot product of two vectors. The contact point P is defined as the intersection of the line through D in the direction of the motion of D, with the contact plane $A\ B\ C$. The relative slip is computed as:

$$g_t = |\boldsymbol{PP}_o|$$

where P_o is the position at which node D first contacted the plane. The normal and tangential stresses are computed as:

$$S_n = k_n \cdot g_n / Area \qquad S_t = k_t \cdot (g_t - g_{t_o}) / Area$$

where Area = area of contact plane $A\ B\ C$, and g_{t_o} is a memory parameter.

The normal stress must be compressive, i.e.,

$$S_n \le 0$$

and the tangential stress must satisfy the Coulomb friction constraint as:

$$|S_t| + S_n \tan\varphi - c \le 0$$

where φ = friction angle, and c = cohesion. A return procedure is used to enforce the inequality when violated.

CONTACT_PLANE

Element_name = CONTACT_PLANE
m, stiff(1, m), stiff(2, m), phi(m), c(m) < m = 1, numat >
< connectivity data >
< terminate with a blank record >.

9.8.1 Element Group Control Information

Must follow the element name (same data record), and define the control parameter as follows:

Note	Variable Name	Type	Default	Description
(1)	Gapping **on** / off	list	[on]	Gapping code
	Friction_load_time	integer	[0]	Friction angle load time function number
	formulation **penalty** augmented_lagrangian	list	[penalty]	Formulation

Notes /
(1) This allows the contact-release option to be deactivated if needed.

9.8.2 Geometric / Material Properties Data (Numat sets)

Note	Variable	Default	Description
	M	[0]	Geometric/material set number
	STIFF(1,M)	[0.0]	Spring constant k_t
	STIFF(2,M)	[0.0]	Spring constant k_n
	PHI(M)	[0.0]	Friction angle (degrees)
	C(M)	[0.0]	Cohesion

9.8.3 Element Nodal Connectivity Data

Consult Chapter 11 for details; for this element NEN = 4, and the nodes are entered in the following order; node A, node B, node C and node D.

9.8.4 Output History Requests

Plots of various element response components may be obtained. Each component required is plotted versus time. Plots of this type are useful in providing quick information concerning the time history behavior of important data. The total number of components to be plotted must equal NOUT, which is defined on the element group control record (see Section 9.8.1).

Note	Variable	Default	Description
(1)	N	[0]	Element number ≥ 1 and ≤ Numel
(2)	NG	[0]	Generation increment ≥ 0
(3)	NTEMP(1)	[0]	Component number ≥ 1 and ≤ 2
	NTEMP(2)	[0]	Component number ≥ 1 and ≤ 2

Notes/
(1) Element components history output data must be input for elements at which the time history of one or more components is to be plotted. Records need not be read in order. *Terminate with a blank record.*

(2) Element components history output data can be generated by employing a two record sequence as follows:

Record 1: L,LG,LTEMP(1),LTEMP(2)
Record 2: N,NG,NTEMP(1),NTEMP(2)

The output time history requests of all elements

L+LG, L+2*LG,...,N-MOD(N-L,LG)

(i.e., less than N) are set equal to those of node L. If LG is blank or zero, no generation takes place between L and N.

(3) Two different component numbers may be plotted as described above. The corresponding component numbers and output labels are:

Component Number	Description	Output Label
1	Normal contact force	F_n
2	Tangential contact force	F_t

Notes . .

Notes . .

Notes . .

9.9 Xfem Crack Element

The element is used to model cracks within the context of xfem (extended finite element) methods. The element current implementation is restricted to 2D problems.

CRACK_Xfem

Element_name = CRACK_Xfem, etc. …
< material data >
< connectivity data >.

9.9.1 Element Control Information

Note	Variable Name	Type	Default	Description
	Element_name	list	[*]	CRACK_Xfem
	Element_type	list	[interface]	Element type
	Element_shape	list	[two_node_line]	Element shape
	Crack_growth on / **off**	list	[off]	Crack growth flag
	Number_of_elements	integer	[*]	Maximum number of elements
	Enrichment_type	list	[none]	Enrichment type
	heaviside			Heaviside only
	heavi_crack_with_tip_1			Heaviside crack with tip 1
	heavi_crack_with_tip_2			Heaviside crack with tip 2
	crack_with_tip_1			Heaviside + asymptotic crack function for tip 1
	crack_with_tip_2			Heaviside + asymptotic crack function for tip 2
	crack			Heaviside + asymptotic crack functions for both tips 1 and 2
	interface_with_tip_1			Heaviside + asymptotic crack function for tip 1
	interface_with_tip_2			Heaviside + asymptotic crack function for tip 2
	interface			Heaviside + asymptotic crack functions for both tips 1 and 2

(cont'd)

(cont'd)

Note	Variable Name	Type	Default	Description
	shear_band joint discontinuities			Tangential discontinuity Tangential + normal
	cohesive_with_tip_1 cohesive_with_tip_2 cohesive			Heaviside + cohesive tip for tip 1 Heaviside + cohesive tip for tip 2 Heaviside + cohesive tip for both tips 1 and 2
	Interface_crack on / **off**	list	[off]	Interface crack flag
	Crack_material on / **off**	list	[off]	Crack material data

9.9.2 Material Properties Data

Note	Variable Name	Type	Default	Description
	Crack_material_model	list	[none]	Title
	Radius	real	[2.5]	Radius for computing stress intensity factors (fraction of h-elmt)
	Critical_energy	real	[0.0]	Critical energy release rate, G_c
	Crack_growth_inc	real	[0.0]	Maximum crack growth increment, da_{max}
	Growth_direction **sigma_theta** straight	list	[*]	Growth direction
(1)	Growth_formula **none** hyperbolic_sin logarithmic_formula hyperbolic_tan	list	[*]	Growth formula
(1)	Threshold_energy	real	[0.0]	Threshold energy release rate G_{th}
(1)	Growth_alpha	real	[0.0]	Growth parameter alpha, α

(cont'd)

(cont'd)

Note	Variable Name	Type	Default	Description
(1)	Growth_beta	real	[0]	Growth parameter beta, β
(2)	Interface_type **perfect** / contact	lst	[*]	Interface type
(2)	Friction_angle	real	[0.0]	Friction angle in degrees
(2)	Normal_stiffness	real	[0.0]	Normal stiffness
(3)	Tangential_stiffness	real	[0.0]	Tangential stiffness
(3)	Number_of_integration	integer	[2]	Number of integration points
(4)	Cohesive_traction	real	[0.0]	Cohesive strength f_u
(4)	Fracture_energy	real	[0.0]	Fracture energy G_F

Notes/

(1) The crack growth velocity is computed as:

- hyperbolic_sin formula:

$$\frac{da}{dt} = \alpha \sinh\left[\beta\left(\frac{G}{G_{th}} - 1\right)\right]$$

- logarithmic formula:

$$if \quad G < G_{th} \qquad \frac{da}{dt} = 0$$

$$if \quad G_{th} \le G < G_c \qquad \frac{da}{dt} = \alpha G^{\beta}$$

$$if \quad G \ge G_c \qquad \frac{da}{dt} = \left(\frac{da}{dt}\right)_{\max}$$

- hyperbolic_tan formula:

$$\frac{da}{dt} = \alpha \tanh\left[\beta\left(\frac{G}{G_{th}} - 1\right)\right]$$

(2) Only applicable to Joint interfaces.

(3) Only applicable to Joint and Shear Band interfaces.

(4) A linear strain softening bridging law is assumed, viz., the critical opening w_e defined as: $w_e = 2G_F/f_u$.

9.9.3 Nodal Connectivity Data

Consult Chapter 11 for details; for this element NEN=2.

References / Bibliography

1. Moës, N., J. Dolbow and T. Belytschko, "A Finite Element Method for Crack Growth without Remeshing," *International Journal for Numerical Methods in Engineering*, Vol. 46, No. 1, (1999), pp. 131-150.
2. Daux, C., N. Moës, J. Dolbow, N. Sukumar and T. Belytschko, "Arbitrary Cracks and Holes with the Extended Finite Element Method," *International Journal for Numerical Methods in Engineering,* Vol. 48, No. 2, (2000), pp. 1741-1760.

Notes . .

Notes . .

Notes . .

Notes . .

9.10 Level Set Element

The element is used to define the initial position of the zero level set curve. The element current implementation is restricted to 2D problems.

LEVEL_XFEM

Element_name = Level_Xfem, etc. …
< material data >
< connectivity data >.

9.10.1 Element Control Information

Note	Variable Name	Type	Default	Description
	Element_name	list	[*]	Level_Xfem
	Element_type	list	[interface]	Element type
	Element_shape	list	[two_node_line]	Element shape
	Level_set_type **signed_distance**	list	[*]	Level set type signed distance function
(1)	Enrichment_type signed_distance	list	[none]	Enrichment type
	Level_set_material	list	[off]	Level set material data

Notes/

(1) Only applicable to Xfem calculations.

9.10.2 Material Properties Data

Note	Variable Name	Type	Default	Description
	Level_set_material_model	list	[none]	Title
	Critical_energy	real	[0.0]	Critical energy release rate, G_c
(1)	Growth_formula **none** / hyperbolic_sin	list	[*]	Growth formula
(1)	Threshold_energy	real	[0.0]	Threshold energy release rate G_{th}
(1)	Growth_V0	real	[0.0]	Growth velocity, V0
(1)	Growth_beta	real	[0.0]	Growth parameter beta, β

Notes/

(1) The velocity is computed as: $V = V0 \sinh\left[\beta\left(\frac{G}{G_{th}} - 1\right)\right]$.

9.10.3 Nodal Connectivity Data

Consult Chapter 11 for details; for this element NEN=2.

Notes . .

Notes . .

9.11 Multi_Point_Constraints

9.11.1 Multi_Point_Constraint

The element is used to impose linear multi nodal point constraint conditions. A typical constraint equation is written as:

$$\sum_i c_i d_j^n = c_o f(t)$$

Where the $d_j^n =$ are the selected nodal point degrees of freedom j at node n. Constraints may originate from several kinds of physical specifications, such as skewed nodal displacement boundary conditions, interface conditions between regions (solid-solid contact, fluid-solid interactions), etc. An augmented lagrangian formulation with penalty regularization is used to satisfy the constraints. Uzawa's algorithm is used to update the lagrange multipliers. A load time function (ltime) can be used to vary c_o with time.

MULTI_POINT_CONSTRAINT

Element_name = MULTI_POINT_CONSTRAINT
m, ltime(m), penalty(m), c0(m), (idof(i,m), c(i,m), i=1,ndof)
< connectivity data >
< terminate with a blank record >.

EXAMPLE

```
Define_element_group   /
        name = "mpc"   /
        element_name = multi_point_constraint   /
        element_type = interface   /
        number_of_coeff_1 =  3   /
        number_of_material_sets = 1

c       m       ltime           penalty   c0          (idof, c1)    (idof, c2)     (idof, c3)
        1       2               1 e+3     1.0         1, 1.0        1, -1.0        2, -1.0

Nodal_connectivity   /
        input_format = list
        1       1       1        5       0
        2       1       6       10       9
```

In this case the element is used to impose the following constraints:

at nodes 1 and 5: $c_1 d_1^1 + c_2 d_1^5 = c_0 f_2(t)$

at nodes 6, 9 and 10: $c_1 d_1^6 + c_2 d_1^{10} + c_3 d_2^9 = c_0 f_2(t)$

References / Bibliography

1. Arrow, K.J., Hurwicz, L. and Uzawa, H., *Studies in Nonlinear Programming*, Stanford University Press, Stanford, 1958.

9.11.1.1 Element Control Information

Note	Variable	Type	Default	Description
	Element_name	list	[*]	Multi_point_constraint
	Element_type	list	[interface]	Element_type
	Element_shape	list	[none]	Element_shape
'				
	Number_of_coeff_1	integer	[max [3, 1 + ndof]]	Number of coefficients c_i*'s*
	Number_of_material_sets	integer	[0.0]	Number_of_material_sets

9.11.1.2 Material Properties (Numat sets)

Note	Variable	Default	Description
	M	[0]	Geometric/material set number
	LTIME(M)	[0]	Load-time function number
	PENALTY(M)	[0.0]	Penalty coefficient
	C0(M)	[0.0]	Coefficient c_0
	IDOF(i,M)	[0]	Degree of freedom number *j*
	C(i,M)	[0.0]	Coefficient c_i

9.11.1.3 Nodal Connectivity Data

Consult Chapter 11 for details; for this element NEN = number_of_coeff_1

9.11.2 Multi_Point_BC

The element is used to impose linear multi nodal point constraint conditions. A typical constraint equation is written as:

$$\left[\sum_i c_i d_j^n \; - c_o \right] f(t) = 0$$

Where the $d_j^n =$ are the selected nodal point degrees of freedom j at node n. Constraints may originate from several kinds of physical specifications, such as skewed nodal displacement boundary conditions, interface conditions between regions (solid-solid contact, fluid-solid interactions), etc. An augmented lagrangian formulation with penalty regularization is used to satisfy the constraints. Uzawa's algorithm is used to update the lagrange multipliers. A load time function (ltime) is used to apply the constraint as a function of time.

MULTI_POINT_BC

Element_name = MULTI_POINT_BC
m, ltime(m), penalty(m), c0(m), (idof(i,m), c(i,m), i=1,ndof)
< connectivity data >
< terminate with a blank record >.

EXAMPLE

```
Define_element_group   /
        name = "mpc"   /
        element_name = multi_point_BC   /
        element_type = interface   /
        number_of_coeff_1 = 3   /
        number_of_material_sets = 1

c       m       ltime           penalty   c0        (idof, c1)    (idof, c2)    (idof, c3)
        1       2               1 e+3     1.0       1, 1.0        1, -1.0       2, -1.0

Nodal_connectivity   /
        input_format = list
        1       1       1       5       0
        2       1       6       10      9
```

In this case the element is used to impose the following constraints:

at nodes 1 and 5: $\left(c_1 d_1^1 + c_2 d_1^5 - c_0 \right) f_2(t) = 0$

at nodes 6, 9 and 10: $\left(c_1 d_1^6 + c_2 d_1^{10} + c_3 d_2^9 - c_0 \right) f_2(t) = 0$

References / Bibliography

1. Arrow, K.J., Hurwicz, L. and Uzawa, H., *Studies in Nonlinear Programming*, Stanford University Press, Stanford, 1958.

9.11.2.1 Element Control Information

Note	Variable	Type	Default	Description
	Element_name	list	[*]	Multi_point_BC
	Element_type	list	[interface]	Element_type
	Element_shape	list	[none]	Element_shape
	Number_of_coeff_1	integer	[max [3, 1 + ndof]]	Number of coefficients c_i's
	Number_of_material_sets	integer	[0.0]	Number_of_material_sets

9.11.2.2 Material Properties (Numat sets)

Note	Variable	Default	Description
	M	[0]	Geometric/material set number
	LTIME(M)	[0]	Load-time function number
	PENALTY(M)	[0.0]	Penalty coefficient
	C0(M)	[0.0]	Coefficient c_0
	IDOF(i,M)	[0]	Degree of freedom number j
	C(i,M)	[0.0]	Coefficient c_i

9.11.2.3 Nodal Connectivity Data

Consult Chapter 11 for details; for this element NEN = number_of_coeff_1

9.12 Nodal Mass / Damping / Stiffness Element

9.12.1 Nodal Mass Element

NODAL_MASS

Element_name = NODAL_MASS
m, (mass(i, m), i = 1, ndof) < m = 1, numat >
< body force data >
< connectivity data >
< terminate with a blank record >.

9.12.1.1 Material Properties Data (Numat sets)

Note	Variable Name	Type	Default	Description
	M	integer	[0]	Geometric/material set number
	MASS(1,M)	real	[0.0]	Nodal mass degree of freedom 1
	MASS(2,M)	real	[0.0]	Nodal mass degree of freedom 2
		.		.
		etc.		.
		.		.
	MASS(NDOF,M)	real	[0.0]	Nodal mass degree of freedom NDOF

9.12.1.2 Body Force Data (units L/T^2)

Note	Variable Name	Type	Default	Description
(1)	b_x1	real	[0.0]	Body force component in the x_1 direction
	b_x2	real	[0.0]	Body force component in the x_2 direction
	b_x3	real	[0.0]	Body force component in the x_3 direction

Notes/
(1) Body force load multipliers are used to define the components of the gravity vector **b** with respect to the global (x_1, x_2, x_3) coordinate system, e.g., in SI units, **b** = {0.0, -9.81, 0.0} for the case x_2 vertical and oriented positively upward, with g = 9.81 m/s^2 and $\rho_W = 10^3$ kg/m^3.

9.12.1.3 Element Nodal Connectivity Data

Consult Chapter 11 for details; for this element NEN = 1.

9.12.2 Nodal Damping Element

NODAL_DAMPING

Element_name = NODAL_DAMPING
m, (damp(i, m), i = 1, ndof) < m = 1, numat >
< connectivity data >
< terminate with a blank record >.

9.12.2.1 Material Properties Data (Numat sets)

Note	Variable Name	Type	Default	Description
	M	integer	[0]	Geometric/material set number
	DAMP(1,M)	real	[0.0]	Nodal damping degree of freedom 1
	DAMP(2,M)	real	[0.0]	Nodal damping degree of freedom 2
		.		.
		etc.		.
		.		.
	DAMP(NDOF,M)	real	[0.0]	Nodal damping degree of freedom NDOF

9.12.2.2 Element Nodal Connectivity Data

Consult Chapter 11 for details; for this element NEN = 1.

9.12.3 Nodal Stiffness Element

NODAL_SPRING

Element_name = NODAL_SPRING
m, (stif(i, m), i = 1, ndof) < m = 1, numat >
< connectivity data >
< terminate with a blank record >.

9.12.3.1 Material Properties Data (Numat sets)

Note	Variable Name	Type	Default	Description
	M	integer	[0]	Material set number
	STIF(1,M)	real	[0.0]	Nodal stiffness value DOF component 1
	STIF(2,M)	real	[0.0]	Nodal stiffness value DOF component 2
	.	.	.	
	etc.	.	.	
	.	.	.	
	STIF(NDOF,M)	real	[0.0]	Nodal stiffness value DOF component NDOF

9.12.3.2 Element Nodal Connectivity Data

Consult Chapter 11 for details; for this element NEN = 1.

Notes . .

Notes . .

Notes . .

9.13 Nodal Reaction Element

The element is used to constrain nodal displacements/rotations in any direction to specified values, to compute support reactions and to provide elastic suppports to nodes. The element is defined by a single directed axis through a specified nodal point, by a linear extensional spring along the axis or by a linear rotational spring about the axis. The element is essentially a spring which can have axial displacement stiffness or rotational stiffness. There is no limit to the number of boundary elements which can be applied at any nodal point to produce the desired effects.

NODAL_REACTION

Element_name = NODAL_REACTION Kinem = kinem, etc...
m, stiff(m), (an(i, m), i = 1, nsd) < m = 1, numat >
  < connectivity data >
< terminate with a blank record >.

9.13.1 Element Group Control Information

Must follow the element name (same data record), and define the control parameters in the following form: KINEM = kinem, etc...

Note	Variable Name	Default	Description
	KINEM	[0]	Boundary condition code = 0 Displacement option = 1 Rotation option

9.13.2 Geometric / Material Properties Data (Numat sets)

Note	Variable	Default	Description
	M	[0]	Geometric/material set number
	STIFF(M)	[0.0]	Spring constant k
	AN(1,M)	[0.0]	Component 1 of direction vector N
	AN(2,M)	[0.0]	Component 2 of direction vector N
	AN(3,M)	[0.0]	Component 3 of direction vector N

9.13.3 Element Nodal Connectivity Data

Consult Chapter 11 for details; for this element NEN = 1.

9.13.4 Element Output History Requests

"On-line" and Calcomp plots of various element response components may be obtained. Each component required is plotted versus time. Plots of this type are useful in providing quick information concerning the time history behavior of important data. The total number of components to be plotted must equal NOUT, which is defined on the element group control record (see Section 9.10.1).

Note	Variable	Default	Description
(1)	N	[0]	Element number ≥ 1 and ≤ Numel
(2)	NG	[0]	Generation increment ≥ 0
(3)	NTEMP(1)	[0]	Component number ≥ 1 and ≤ 2
	NTEMP(2)	[0]	Component number ≥ 1 and ≤ 2

Notes/
(1) Element components history output data must be input for elements at which the time history of one or more components is to be plotted. Records need not be read in order. *Terminate with a blank record.*

(2) Element components history output data can be generated by employing a two record sequence as follows:

Record 1: L,LG,LTEMP(1),LTEMP(2)
Record 2: N,NG,NTEMP(1),NTEMP(2)

The output time history requests of all elements

L+LG, L+2*LG,..., N-MOD(N-L,LG)

(i.e., less than N) are set equal to those of node L. If LG is blank or zero, no generation takes place between L and N.

(3) Two different component numbers may be plotted as described above. The corresponding component numbers and output labels are:

Component Number	Description	Output Label
1	Displacement /Rotation	DELT
2	Force/Moment	FORC

Notes . .

9.14 Nodal Link Element

The link element is used to connect two nodal points either in translations or rotations in any direction. The element is defined by two nodes and a single directed axis, by a linear/nonlinear extensional stiffness and damping along the axis, or by a linear/nonlinear rotational stiffness and damping about the axis. The element is essentially a spring or damper which can have either axial or rotational stiffness and damping. There is no limit to the number of links which can be established between two nodal points to produce the desired effects.

By default a linear spring is assumed. If nonlinear, a piecewise linear behavior is assumed. The nonlinear spring capabilities of the element are illustrated in Figure 9.14.1a.

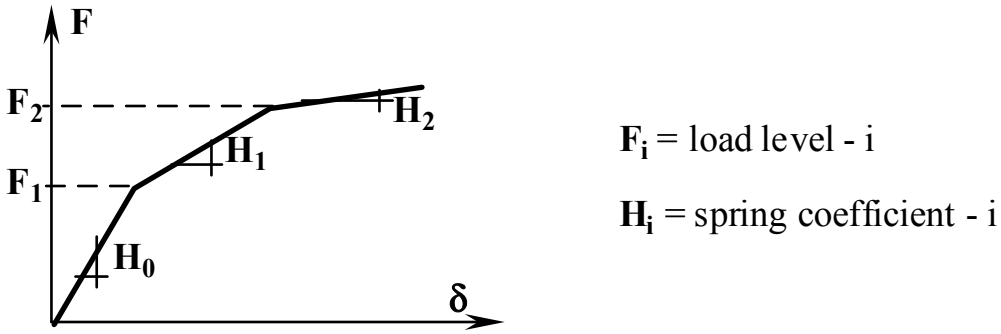


Figure 9.14.1a Nonlinear Link Element

where F = force or moment, and δ = displacement or rotation. The nonlinear curve is assumed piecewise linear and is prescribed by defining various load_levels F_i , and corresponding spring_coefficients H_i up to 5 load_levels and corresponding spring coefficients can be used. The element may be prescribed to be elastic or hysteretic, as shown in Figure 9.14.1b.

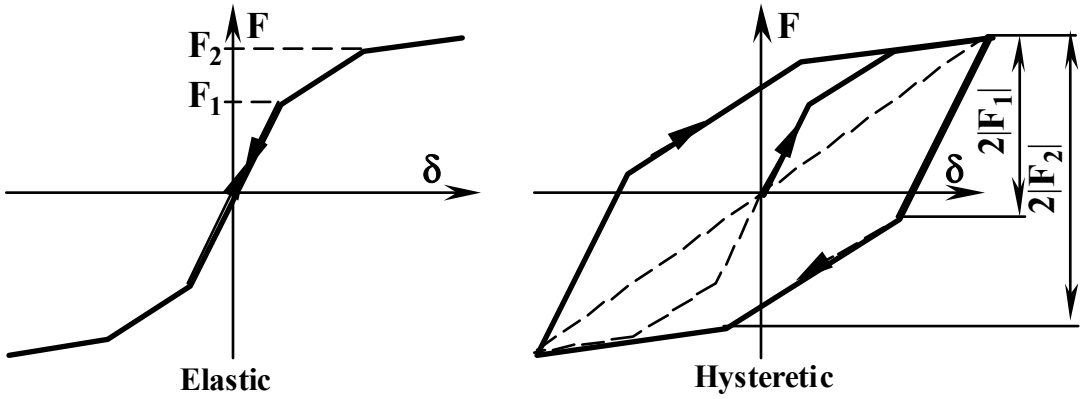


Figure 9.14.1b Nonlinear Link Element

NODAL_LINK

Element_name = NODAL_LINK Kinem = kinem, etc...
< material data >
< output requests >
< connectivity data >

9.14.1 Element Group Control Information

Must follow the element name (same data record), and define the control parameters in the following form: KINEM = kinem, etc...

Note	Variable Name	Type	Default	Description
	Kinem	integer	[0]	Link condition code = 0 Translation = 1 Rotation
	Number_of_material_sets	integer	[1]	Number of material sets
	File_name	string	[none]	File name (optional). Name must be enclosed in quotation marks.
	Input_format **keywords** / list	list	[*]	Input format

9.14.2 Geometric / Material Properties Data (Numat sets)

Note	Variable Name	Type	Default	Description
• *Keywords Read Method*				
	Material_set_number	integer	[1]	Material set number □ Numat
	Modulus_coefficient	real	[0.0]	Spring coefficient H_0
	Damping_coefficient	real	[0.0]	Damping coefficient *c*
(1)	Damping_exponent	real	[0.0]	Damping exponent α
	Reference_axis			
	n_x	real	[0.0]	Component in x_direction
	n_y	real	[0.0]	Component in y_direction
	n_z	real	[0.0]	Component in z_direction
	Material_type **elastic** / hysteretic	string	[elastic]	Material type selection
	Load_level_i	real	[0.0]	Load level i (i ≤ 5)
	Spring_coefficient_i	real	[0.0]	Spring coefficient i (i ≤ 5)
	Tensile_strength	real	[0.0]	Tensile failure strength (only active if > 0)
	Compressive_strength	real	[0.0]	Compressive failure strength (only active if < 0)

• *List Read Method*

Geometric data must follow in the form:

< n, spring (n), damp (n), n_x (n), n_y (n), n_z (n),
mat_type (n), load_level (i,n), spring_coeff (i,n), i = 1, 5 >
< terminate with a blank record >.

Notes/

(1) The damping coefficient is computed as:

$$C = c|\mathbf{v}|^{\alpha}$$

where $|\mathbf{v}|$ = relative velocity between node 1 and node 2; *c* = damping_coefficient and α = damping_exponent.

EXAMPLE

```
DEFINE_ELEMENT_GROUP /
    name = "group_name" /
    element_type = nodal /
    element_name = nodal_link /
```

```
number_of_output_sets = 2 /
kinem= 1 /
number_of_material_sets = 1

    material_set_number = 1 \
        modulus_coeff = 1.3+4 \
        n_x=0.00  n_y=0.00  n_z=1.00 \
            material_type = hysteretic \
            load_level_1 = 15.0 \
            spring_coeff_1 = 5.e+3

Nodal_connectivity
    1       1       1       2

Field_output
    1       0       1       2
```

9.14.3 Element Nodal Connectivity Data

Consult Section 11 for details; for this element NEN = 2.

9.14.4 Output History Requests

FIELD_OUTPUT

FIELD_OUTPUT n, ng, ntemp(1), ntemp(2), etc...
< etc..., terminate with a blank record >.

Plots of various element response components may be obtained. Each component requested is plotted versus time. Plots of this type are useful in providing quick information concerning the time history behavior of important data. The total number of components to be plotted must equal Number_output_sets, which is defined on the element group control command (see Section 9.1).

Note	Variable	Default	Description
(1)	N	[0]	Element number ≥ 1 and ≤ Numel
(2)	NG	[0]	Generation increment ≥ 0
(3)	NTEMP(1)	[0]	Component number ≥ 1 and ≤ 2
	NTEMP(2)	[0]	Component number ≥ 1 and ≤ 2

Notes/
(1) Element components history output data must be input for elements at which the time history of one or more components is to be plotted. Records need not be read in order. *Terminate with a blank record.*

(2) Element components history output data can be generated by employing a two record sequence as follows:

Record 1: L,LG,LTEMP(1),LTEMP(2)
Record 2: N,NG,NTEMP(1),NTEMP(2)

The output time history requests of all elements

L+LG, L+2*LG,..., N-MOD(N-L,LG)

(i.e., less than N) are set equal to those of node L. If LG is blank or zero, no generation takes place between L and N.

(3) Two different component numbers may be plotted as described above. The corresponding component numbers and output labels are:

Component Number	Description	Output Label
1	Displacement /Rotation	DELT
2	Force/Moment	FORC

Notes . .

9.15 Selective Nodal Penalty Element

The nodal penalty element is used to restrain a nodal value at a particular node, in a particular direction, to remain positive. The element is used in free-surface searches (viz., unconfined flow problems). Nodal penalty elements have no effect on the size of the stiffness matrix.

NODAL_PENALTY

Element_name = NODAL_PENALTY
m, (pen(i, m), i = 1, ndof) < m = 1, numat >
< connectivity data >
< terminate with a blank record >.

9.15.1 Material Properties Cards (Numat sets)

Note	Variable	Default	Description
	M	[0]	Material set number
	PEN(1,M)	[0.0]	Nodal value DOF component 1
	PEN(2,M)	[0.0]	Nodal value DOF component 2
	.	.	.
	etc	.	.
	.	.	.
	PEN(NDOF,M)	[0.0]	Nodal value DOF component NDOF

9.15.2 Element Nodal Connectivity Data

Consult Chapter 11 for details; for this element NEN = 1.

Notes . .

9.16 Transmitting Nodal Boundary Element

The element is used to provide a transmitting nodal boundary for incident propagating motions, such as occur in seismic response calculation. The boundary is frequency independent, and is local in space and time (see Note at end of this chapter for more details). It is exact for vertically propagating wave motions and linear systems only. At the boundary:

$$\tau(t) = \rho C(2sI_{,t} - u_{,t})$$

where t = time, ρ = mass density of underlying medium, C = wave speed, $I = I(t)$ = incident motion, $u = u(t)$ = associated displacement motion at the node, and s = multiplier. The incident motion $I(t)$ is defined by a corresponding load-time function.

NODAL_TRANSMITTING

Element_name = NODAL_TRANSMITTING etc...
< material data >
< output requests >
< connectivity data >

9.16.1 Element Group Control Information

Must follow the element name (same data record), and define the following:

Note	Variable Name	Type	Default	Description
	Number_of_material_sets	integer	[1]	Number of material sets, Numat
	File_name	string	[none]	File name (optional). Name must be enclosed in quotation marks.
	Input_format **keywords** / list	list	[*]	Input format

9.16.2 Geometric / Material Properties Data (Numat sets)

Note	Variable Name	Type	Default	Description
• *Keywords Read Method*				
	Material_set_number	integer	[0]	Material set number ≤ Numat
	Incident_motion_definition **displacement** velocity acceleration	list	[*]	Incident motion definition
	Incident_motion_load_time	integer	[0]	Incident motion load-time function number
	Motion_multiplier_i	real	[0.0]	Motion multiplier for degree of freedom i $(1 \le i \le 6)$.
	Rhoc_i	real	[0.0]	ρC for degree of freedom i $(1 \le i \le 6)$.

• *List Read Method*

Geometric / material data must follow in the form:

< m, $ltime(m)$, $(s(i, m), i = 1, ndof)$ >

< $(rhoc(i, m), i = 1, ndof)$ >

< terminate with a blank record >.

EXAMPLE

```
Define_Element_Group  /
    name = "Group_3"  /
    element_type = nodal  /
    element-shape = one_node  /
    number_of_material_sets = 1

    material_set_number = 1  /
        incident_motion_definition = displacement  /
        incident_motion_load_time = 1  /
        motion_multiplier_1 = 1.00  /
        rhoc_1  = 105.175e6

NODAL_CONNECTIVITY  etc…
```

9.16.3 Element Nodal Connectivity Data

Consult Chapter 11 for details; for this element NEN = 1.

Note/

Transmitting Boundary

1. Introduction

The application of finite elements to the solution of problems involving the propagation of waves requires the development of special boundary conditions referred to as transmitting, non-reflecting, silent or energy-absorbing boundaries. These boundary conditions are required to use at the boundary of the necessarily finite mesh to simulate the infinite extent of the domain. For instance, when an infinite domain in the vertical direction is modeled by a finite mesh, there is danger that waves reflected from the free-surface will be reflected back off the artificial bottom boundary and cause errors in the response calculations, unless special boundary conditions can be imposed at the base of the column. In the following, a rigorous formulation of an appropriate boundary condition is presented. The proposed boundary condition is frequency independent, and is local in space and time. It is exact for linear systems only, and therefore requires that the boundary be placed at a sufficiently large distance such that the response be linear at that distance.

Seismic site response calculations are usually performed for a given seismic input prescribed in the form of an acceleration time history to be applied at the base of the soil column. As discussed hereafter, the implementation of an appropriate boundary condition at the base of the soil column requires detailed knowledge of the nature of the prescribed seismic input, viz. whether it corresponds to an incident vertically propagating motion or is the sum of an incident and a reflected motion.

The features of one-dimensional wave propagation in a semi-infinite system are first reviewed before the boundary condition is developed.

2. One-Dimensional Vertical Wave Propagation

For the purpose of illustrating the features of the boundary formulation, the vertical propagation of shear waves is considered. The equation of motion may be expressed as:

$$\rho u_{,tt} = G u_{,xx} \tag{1}$$

where a comma is used to indicate partial differentiation; ρ = mass density; G = shear modulus; u = horizontal displacement; t = time; and x = depth coordinate, with the x-coordinate assumed oriented upwards positively. The fundamental solution of Eq. 1 can be expressed as:

$$u(x,\ t) = I\left(t - \frac{x}{C}\right) + R\left(t + \frac{x}{C}\right) \tag{2}$$

where

$$C = \sqrt{\frac{G}{\rho}} \tag{3}$$

and I and R are two arbitrary functions of their arguments: $I\left(t - \frac{x}{C}\right)$ represents a wave motion propagating upwards in the positive x-direction with the velocity C, and is referred to as the

incident motion; $R\left(t+\frac{x}{C}\right)$ presents a wave motion propagating downwards in the negative x-direction with the velocity *C*, and is referred to as the *reflected motion*. The following two identities apply:

$$I_{,x}+\frac{1}{C}I_{,t}=0 \tag{4a}$$

$$R_{,x}-\frac{1}{C}R_{,t}=0 \tag{4b}$$

and therefore, if one differentiates Eq. 2 with respect to x and t in turn:

$$u_{,x}=\frac{1}{C}\left(-I_{,t}+R_{,t}\right) \tag{5}$$

$$u_{,t}=I_{,t}+R_{,t} \tag{6}$$

The shear stress $\tau(x,\ t)$ can therefore be expressed as

$$\tau(x,t)=Gu_{,x}=\rho C\left(-I_{,t}+R_{,t}\right) \tag{7}$$

and upon elimination of $R_{,t}$ the following relation is obtained:

$$\tau(x,t)=\rho C\left(u_{,t}-2I_{,t}\right) \tag{8}$$

At this stage it is instructive to study the total wave pattern when an incident wave motion $I\left(t-\frac{x}{C}\right)$ encounters an artificial boundary at $x=h$. Three extreme cases can be considered as follows:

2.1 The boundary at $x=h$ is fixed.

Setting $u(h,\ t)=0$ in Eq. 2 leads to:

$$R\left(t+\frac{x-h}{C}\right)=-I\left(t-\frac{x-h}{C}\right) \tag{9}$$

resulting in the total wave motion:

$$u(x,t)=I\left(t-\frac{x-h}{C}\right)-I\left(t+\frac{x-h}{C}\right) \tag{10}$$

Therefore, at a fixed boundary, the incident wave is reflected back with the same shape but opposite sign.

2.2 The boundary at $x = h$ is free.

Setting $\tau(h,\ t) = 0$ in Eq. 7 leads to:

$$R\left(t + \frac{x-h}{C}\right) = I\left(t - \frac{x-h}{C}\right) \tag{11}$$

resulting in the total wave motion:

$$u(x,\ t) = I\left(t - \frac{x-h}{C}\right) + I\left(t + \frac{x-h}{C}\right) \tag{12}$$

Therefore, at a free boundary, the incident wave is reflected back with the same shape and the same sign.

2.3 The boundary at $x = h$ is silent.

Selecting Eq. 4a which is identically satisfied for *I* as the boundary condition for u at $x = h$

$$\left(u_{,x} + \frac{1}{C}u_{,t}\right)\Big|_{x\,=\,h} = 0 \tag{13}$$

results in $R = 0$. Eq. 13 is called the radiation condition. It is obtained by selecting:

$$\tau(h,\ t) = -\rho C u_{,t}\big|_{x\,=\,h} \tag{14}$$

When the incident wave *I* encounters that boundary, it passes through it without modification and continues propagating towards $x = +\infty$. No reflected wave *R*, which would propagate back in the negative x-direction can arise.

3. Semi-Infinite Column

Consider the situation shown in Fig. 9.13.1. An incident vertically propagating wave *I* (coming from infinity) arrives at the site, and it is sought to compute the site response for this incident motion. The finite element mesh has been selected to extend down to the depth h, and an appropriate boundary condition at the base of the soil column is sought to simulate the infinite extend of the soil domain in the vertical downward direction. For the purpose of illustration, it is assumed that the site consists in general, of two homogeneous deposits with material properties as follows:

- $(\rho,\ C)$ above the base of the soil column: $0 \le x \le h$
- $(\rho_\infty,\ C_\infty)$ below the base of the soil column: $-\infty < x < 0$

In order to separate the influence of the incident wave from the reflected wave on the site response, it is assumed that the incident motion disturbance spans over a duration $\bar{t}$ with:

$$0 < \bar{t} \le \frac{h}{C}$$

and that it reaches the location $x = 0$ at time $t = 0$. Several cases are considered as follows:

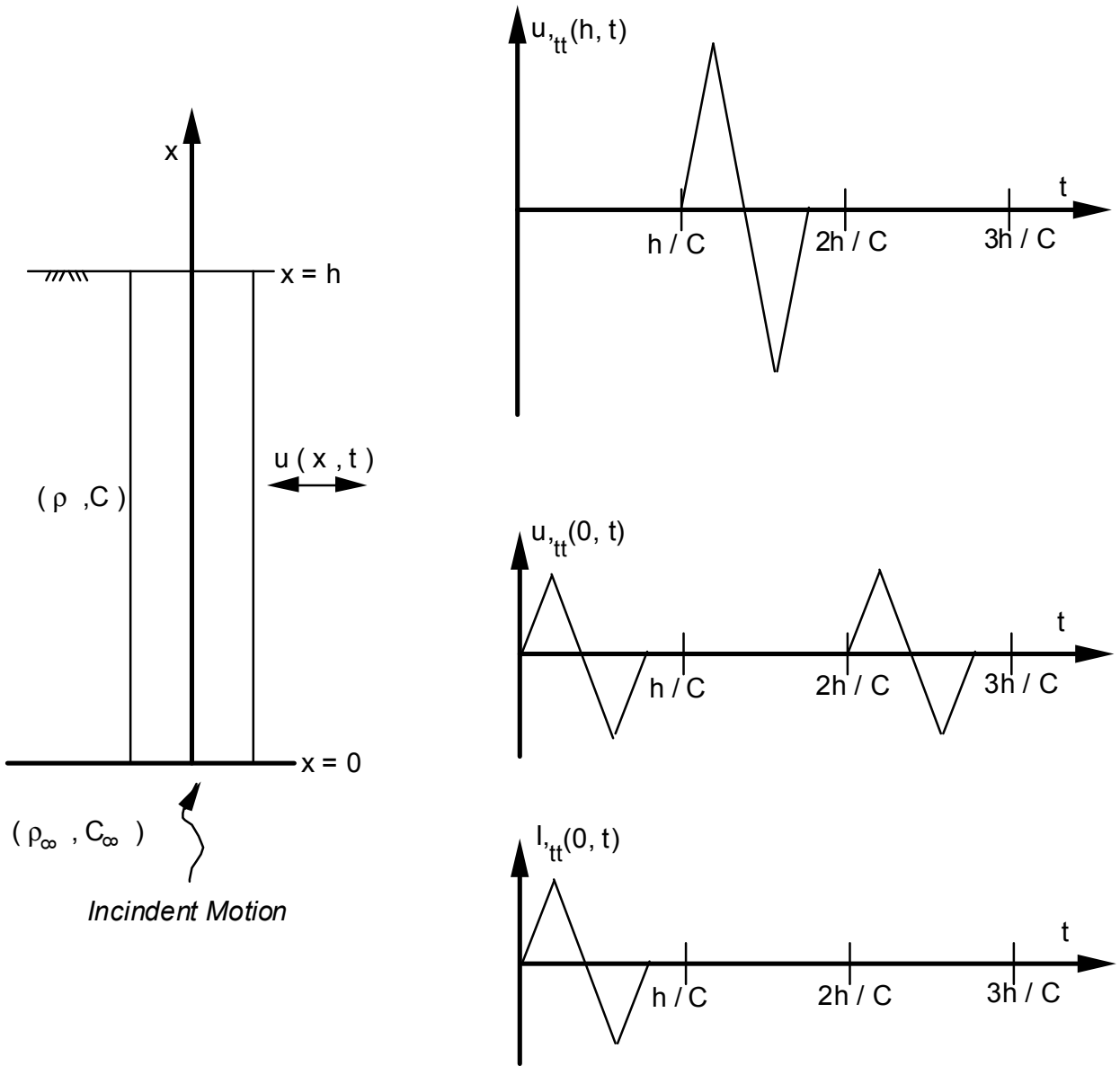


Figure 9.16.1 Semi-Infinite Layered Soil Profile

3.1 Case 1: Homogeneous semi-infinite deposit (viz., $\rho_\infty = \rho; C_\infty = C$):

In that case the incident vertically propagating wave arriving at $x = 0$ at time $t = 0$, will reach the free surface $x = h$ at time $t = \frac{h}{C}$, will be reflected back from the free surface with the same shape and sign, and must cross the boundary at $x = 0$ (at times $t \geq \frac{2h}{C}$) without any further modification and continue propagating back towards infinity. The resulting motions are as follows:

- at $x = 0$

$$u(0,t) = I(t) + I\left(t - \frac{2h}{C}\right) H\left(t - \frac{2h}{C}\right) \tag{15}$$

- at $x = h$

$$u(h,t) = 2I\left(t - \frac{h}{C}\right) H\left(t - \frac{h}{C}\right) \tag{16}$$

where *H* is the Heaviside function. this is illustrated in Fig. 9.13.1. The desired response in the finite soil column can be achieved by prescribing at the base of the soil column either the total motion or the incident part of the motion only, as follows:

▷ *Prescribed motion (fixed base case).* In that case the base input motion must be made up of the incident and reflected motions to reproduce the specified site response as

$$u(0,t) = I(t) + I\left(t - \frac{2h}{C}\right) H\left(t - \frac{2h}{C}\right) \tag{17}$$

The first part of the input corresponding to $I(t)$ in Eq. 17 will propagate towards the surface and reproduce the prescribed surface motion. It will then be reflected back off the free surface towards the fixed base where it will be reflected again with a negative amplitude:

$$-I\left(t - \frac{2h}{C}\right) H\left(t - \frac{2h}{C}\right)$$

This reflected wave is canceled exactly by the second part of the input motion in Eq. 17 thereby preventing any further propagation of waves towards the surface. In other words, the incident wave $I(t)$ produces the surface motion and the reflected wave cancels the reflection from the rigid base.

Remark: The total motion is the one computed in standard deconvolution procedures implemented in computer programs such as SHAKE (1972).

▷ *Prescribed traction (non-reflecting case).* From Eq. 8 the stress in the semi-infinite soil deposit at location $x = 0$ can be expressed directly in terms of the motion at the location and the incident wave motion. Therefore, it suffices to apply at the artificial boundary $x = 0$ the traction:

$$\tau(0,t) = \rho C\left(2I_{,t} - u_{,t}\right)\Big|_{(0,t)} \tag{18}$$

In that case, the incident input motion is absorbed exactly at the base after reflection from the surface. Eq. 18 is the most general boundary condition since it only requires knowledge of the incident motion.

3.2 Case 2: Non-homogeneous semi-infinite deposit.

In that case only the incident motion is known as it arrives at location $x = 0$. In order to compute the site response for this incident motion, accounting for the effects of ensuing reflections (or no reflections if $C_\infty = C$ and $\rho_\infty = \rho$) at the boundary $x = 0$, one must prescribe the input at the base of the finite soil column in terms of prescribed tractions as:

$$\tau(0,t) = \rho_\infty C_\infty\left(2I_{,t} - u_{,t}\right)\Big|_{(0,t)} \tag{19}$$

This will ensure proper simulation of the infinite extend of the soil domain in the downward direction.

References / Bibliography

1. Schnabel, P.B., J. Lysmer and H.B. Seed, "SHAKE: A Computer Program for Earthquake Response Analysis of Horizontally Layered Sites", *Report No. EERC 72-12,* University of California, Berkeley, (1972).

Notes . .

9.17 Free-Field Element

The element is used to input the free-field excitation of a linear soil system to a (nonlinear) soil-structure interacting system. The method permits the free-field excitation to be specified within the region of computation, arbitrarily close to a zone that includes the (possibly nonlinear) structure and local subgrade and backfill, thus eliminating the need to transmit the seismic excitation through artificial boundaries. The method and implementation are based on the work reported by Bielak and coworkers (see e.g. Bielak and Christiano [1984]; Cremonini et al [1988]). The element requires that inside and outside free-field nodes data be defined (see Section 7.5.4).

References / Bibliography

1. Bielak, J. and Christiano, P., "On the Effective Seismic Input for Non-Linear Soil-Structure Interaction Systems," *Earthquake Engineering and Structural Dynamics*, Vol. 12 (1984), 107-119.

2. Cremonini, M, Christiano, P. and Bielak, J., "Implementation of Effective Seismic Input for Soil Structure Interaction Systems," *Earthquake Engineering and Structural Dynamics,* Vol. 16 (1988), 615-625.

9.17.1 Element Control Information

Note	Variable Name	Type	Default	Description
	Element_name	list	[QDC_solid]	Element name
	Element_shape eight_node_brick twenty_node_brick six_node_wedge fifteen_node_wedge four_node_tetra ten_node_tetra four_node_quad eight_node_quad nine_node_quad three_node_tri six_node_tri	list	[none]	Element shape (see Fig. 9.2.0)
	Analysis_type **plane** three_dimensional	list	[plane]	Analysis type: Two-dimensional / plane strain Three-dimensional
	Numerical_integration **full** / reduced	list	[full]	Numerical integration option: If reduced: one-point Gaussian quadrature
	Strain_displacement **standard** bbar_select bbar_mean	list	[*]	Strain-displacement option: Standard formulation Selective-reduced integration Mean-dilatational formulation
	Mass_type **lumped** / consistent	list	[lumped]	Mass type If lumped: row-sum technique
(1)	Free_field **on** / off	list	[on]	Free field option
	Number_of_stress_points	integer	[*]	Number of stress points option: = 1, one stress point Default:one per integration point

(cont'd)

(cont'd)

Note	Variable Name	Type	Default	Description
	Implicit_explicit_type	list	[*]	Implicit/explicit option:
	implicit			Implicit element
	explicit			Explicit element
	implicit_explicit			Implicit-explicit element
	Body_force_load_time	integer	[0]	Body force load-time function number

EXAMPLE

```
Element_Group  /
    name = "Group 1"  /
    element_type = continuum  /
    element_shape = four_node_quad  /
    number_of_material_sets = 1  /
    element_name = QDC_solid  /
        free_field = on  /
        strain_displacement = bbar_mean  /
        number_of_output_sets = 1

Stress_Model  /
    material_type = linear  /
    material_name = linear_elastic
        material_set_number = 1  /
        youngs_modulus = 1.0E4  /
        poissons_ratio = 0.25  /
        solid_mass_density = 2.0  /
        fluid_mass_density = 1.0  /
        porosity = 0.30

Body_force      b_x1 = 0.00      b_x2 = 0.00      h = 0.00

Field_output
    58          0          17

Nodal_connectivity      etc...
```

Notes/
(1) This option requires that inside and outside free_field nodes data be defined (see Section 7.5.4).

9.17.2 Material Data

Material data must be defined for the element group. Consult Chapter 10 for the required input of the individual material models. Note that for this element, it is required that a stress_material_model be prescribed.

9.17.3 Body Force Data (□ □units: L/T^2)

BODY_FORCE

BODY_FORCE b_x1 = b(1) , ... etc

Note	Variable Name	Type	Default	Description
(1)	b_x1	real	[0.0]	Body force component in the x_1 direction
	b_x2	real	[0.0]	Body force component in the x_2 direction
	b_x3	real	[0.0]	Body force component in the x_3 direction

Notes/
(1) Body force load multipliers are used to define the components of the gravity vector **b** with respect to the global (x_1, x_2, x_3) coordinate system, e.g., in SI units, **b** = {0.0, -9.81, 0.0} for the case x_2 vertical and oriented positively upward, with g = 9.81 m/s^2 and $\rho_W = 10^3$ kg/m^3.

9.17.4 Nodal Connectivity Data

Consult Chapter 11 for details. For this element NEN = number of nodes used to define the element (see Fig. 9.2.0.1). For instance: NEN = 4 in two-dimension (four_node_quad), and NEN = 8 in three-dimension (eight_node_brick).

9.17.5 Output History Requests

FIELD_OUTPUT

FIELD_OUTPUT
n, ng, ntemp(1), ntemp(2), etc...
< etc..., terminate with a blank record >.

Plots of various element response components may be obtained. Each component requested is plotted versus time. Plots of this type are useful in providing quick information concerning the time history behavior of important data. The total number of components to be plotted must equal Number_output_sets, which is defined on the element group control command (see Section 9.2.0.1).

Note	Variable	Default	Description
(1)	N	[0]	Element number ≥ 1 and ≤ NUMEL
(2)	NG	[0]	Generation increment ≥ 0
(3)	NTEMP(1)	[0]	Component number ≥ 1 and ≤ NCOMP
	NTEMP(2)	[0]	Component number ≥ 1 and ≤ NCOMP
	etc.	.	.
	NTEMP(8)	[0]	Component number ≥ 1 and ≤ NCOMP

Notes/
(1) Element components history output data must be input for elements at which the time history of one or more components is to be plotted. *Terminate with a blank record.*

(2) Element components history output data can be generated by employing a two record sequence as follows:

Record 1: L, LG, LTEMP(1),..., LTEMP(8)
Record 2: N, NG, NTEMP(1),..., NTEMP(8)

The output time history requests of all elements:

L+LG, L+2*LG,..., N-MOD(N-L,LG)

(i.e., less than N) are set equal to those of element L. If LG is zero, no generation takes place between L and N.

(2) Up to three (1D case), seventeen (2D case) and nineteen (3D case) different component numbers may be plotted. The corresponding component numbers and output labels are as follows.

Notes from 9.17.5 (cont'd)

Table 9.17.5

One Dimensional Kinematics (NCOMP=3)

Component Number	Description	Output Label
1	Normal stress 11	S11
2	Strain 11	E11
3	Fluid pressure/Temperature	PF

Two Dimensional Kinematics (NCOMP=17)

Component Number	Description	Output Label
1	Normal stress 11	S11
2	Normal stress 22	S22
3	Normal stress 33	S33
4	Shear stress 12	S12
5	Principal stress 1	PS1
6	Principal stress 2	PS2
7	Shear stress	PTAU
8	Stress angle (between PS1 and X1)	SANG
9	Strain 11	E11
10	Strain 22	E22
11	Strain 33	E33
12	Engineering shear strain 12	G12
13	Principal strain 1	PE1
14	Principal strain 2	PE2
15	Engineering shear strain	PGAM
16	Strain angle (between PE1 and X1)	EANG
17	Fluid pressure/Temperature	PF

Notes from 9.17.5 (cont'd)

Three Dimensional Kinematics (NCOMP=19)

Component Number	Description	Output Label
1	Normal stress 11	S11
2	Normal stress 22	S22
3	Normal stress 33	S33
4	Shear stress 12	S12
5	Shear stress 23	S23
6	Shear stress 31	S31
7	Principal stress 1	PS1
8	Principal stress 2	PS2
9	Principal stress 3	PS3
10	Strain 11	E11
11	Strain 22	E22
12	Strain 33	E33
13	Engineering shear strain 12	G12
14	Engineering shear strain 23	G23
15	Engineering shear strain 31	G31
16	Principal strain 1	PE1
17	Principal strain 2	PE2
18	Principal strain 3	PE3
19	Fluid pressure/Temperature	PF

Notes . .

Notes . .

Notes . .

10.0 MATERIAL MODELS

A material model is used to describe the material behavior in a region of the finite element mesh. Each group of elements in the finite element mesh requires a material model. The material models may be specified within each region or element group, or may be defined using the command DEFINE_MATERIAL_MODEL. Each material model can then be assigned a name, and the model can be specified in one or more regions by its name (see Section 9.1). Up to Numeg material models can be defined within a problem. Only the material models specified by their name in one or more regions/group of elements will be activated.

10.0 Material Control Information

10.0.1 Define a Material Model

DEFINE_MATERIAL_MODEL

DEFINE_MATERIAL_MODEL name = "<string>" , etc...

Define a material model to be used within a region and/or element group.

Note	Variable Name	Type	Default	Description
	Name	list	[none]	Descriptive name to be used to identify the material model. Name must be enclosed in quotation marks.
	Number_of_material_sets	integer	[1]	Number of material sets
	File_name	string	[none]	File name (optional). Name must be enclosed in quotation marks.
	Input_format **keywords** / list	list	[*]	Input format

Material data must then be defined for the material model. Consult Chapter 10 for the required input for the individual material models.

EXAMPLE

```
Define_material_model  /
    name = "dummy"  /
    number_of_material_sets = 1

    Stress_model  /
        material_name = linear_elastic
            material_set_number = 1  /
            youngs_modulus = 1.0e4  /
            poissons_ratio = 0.25  /
            solid_mass_density = 2.0  /
            fluid_mass_density = 1.0  /
            porosity = 0.30

    Scalar_diffusion_model  /
        material_name = scalar_diffusion
            material_set_number = 1  /
            compressibility = 1.0e-6  /
            permeability  /
                type = isotropic  /
                    k_11 = 1.60e-3
```

10.0.2 Stress Models

STRESS_MODEL

	STRESS_MODEL file_name = "<string>" , etc...		
	This command must precede the material model data.		

Note	Variable Name	Type	Default	Description
	Material_type **linear** / nonlinear	list	[linear]	Material type
	Material_name linear_elastic orthotropic_elastic hyperelastic mcreep mises Drucker_prager Matsuoka cap multi_yield Ishihara Phillips stress_id newtonian_fluid heat_conduction scalar_diffusion electric	list	[none]	Material name: See Section 10.1 See Section 10.2 See Section 10.3 See Section 10.4 See Section 10.5 See Section 10.6 See Section 10.7 See Section 10.8 See Section 10.9 See Section 10.10 See Section 10.11 See Section 10.12 See Section 10.13 See Section 10.14 See Section 10.15 See Section 10.16
(1)	File_name	string	[none]	File name (optional). Name must be enclosed in quotation marks.
	Input_format **keywords** / list	list	[*]	Select input format option

Notes/
(1) This option allows the material data to be read in from another file.

10.0.3 Heat Conduction Models

HEAT_CONDUCTION_MODEL

HEAT_CONDUCTION_MODEL file_name = "<string>" , etc...

This command must precede the material model data.

Note	Variable Name	Type	Default	Description
	Material_type **linear** / nonlinear	list	[linear]	Material type
	Material_name heat_conduction	list	[none]	Material name: See Section 10.12
(1)	File_name	string	[none]	File name (optional). Name must be enclosed in quotation marks
(2)	Number_of_phases	integer	[0]	Number of fluid phases
	Input_format **keywords** / list	list	[*]	Select input format option

Notes/
(1) This option allows the material data to be read in from another file.

(2) Only applicable to multi_phase fluid flow problems (see Sections 9.2.0.15 and 9.2.0.16)

10.0.4 Scalar Diffusion Models

SCALAR_DIFFUSION_MODEL

SCALAR_DIFFUSION_MODEL	file_name = "<string>" , etc...

This command must precede the material model data.

Note	Variable Name	Type	Default	Description
	Material_type **linear** / nonlinear	list	[linear]	Material type
	Material_name scalar_diffusion	list	[none]	Material name: See Section 10.16
(1)	File_name	string	[none]	File name (optional). Name must be enclosed in quotation marks.
(2)	Number_of_phases	integer	[1]	Number of fluid phases
(2)	Number_of_components	integer	[0]	Number of components
(2)	Components_names CO2 NaCl CH4 H2S H2O	string	[none]	Components names. Name(s) must be enclosed in quotation marks.
(2)	Max_number _of_data_points	integer	[0]	Maximum number of data points used to define the relative permeability and capillary pressure curves versus degree of saturation.
	Input_format **keywords** / list	list	[*]	Select input format option

Notes/
(1) This option allows the material data to be read in from another file.

(2) Only applicable to multi-phase fluid flow problems (see Sections 9.2.0.10 and 9.2.0.11).

10.0.5 Electric Models

ELECTRIC_MODEL

ELECTRIC_MODEL file_name = "<string>" , etc...

This command must precede the material model data.

Note	Variable Name	Type	Default	Description
	Material_type **linear** / nonlinear	list	[linear]	Material type
	Material_name electric	list	[none]	Material name: See Section 10.14
(1)	File_name	string	[none]	File name (optional). Name must be enclosed in quotation marks.
	Input_format **keywords** / list	list	[*]	Select input format option

Notes/
(1) This option allows the material data to be read in from another file.

10.1 Stress_Model: Linear Isotropic Elasticity Model

LINEAR_ELASTIC

Material_name = LINEAR_ELASTIC
Material_set_number = mset , etc...

Only two material constants are needed to fully define the linear isotropic elastic model (e.g., Young's modulus and Poisson's ratio.)

Note	Variable Name	Type	Default	Description
• *Keywords Read Method*				
	Material_set_number	integer	[1]	Material set number ≤ Numat
	Elastic_case **incremental** / total	list	[*]	Elastic case
	Plane_stress on / **off**	list	[off]	Plane stress option
	Mass_density	real	[0.0]	Mass density ρ
	Youngs_modulus	real	[0.0]	Young's modulus E
(1)	Poissons_ratio	real	[0.0]	Poisson's ratio ν
	Shear_modulus	real	[0.0]	Shear modulus G
	Bulk_modulus	real	[0.0]	Bulk modulus B
(2)	Mass_damping	real	[0.0]	Rayleigh mass damping coefficient
(2)	Stiffness_damping	real	[0.0]	Rayleigh stiffness damping coefficient
(3)	Solid_mass_density	real	[0.0]	Mass density (Solid Phase) ρ_s
(3)	Fluid_mass_density	real	[0.0]	Mass density (Fluid Phase) ρ_w
(3)	Fluid_bulk_modulus	real	[0.0]	Fluid bulk modulus λ^w
(3)	Porosity	real	[0.0]	Porosity n^w
(4)	Viscoelastic_data	string	[none]	Viscoelastic data
	relaxation_time	real	[0.0]	Relaxation time
	relaxation_bulk_modulus	real	[0.0]	Relaxation bulk modulus value
	relaxation_shear_modulus	real	[0.0]	Relaxation shear modulus value

(cont'd)

(cont'd)

Note	Variable Name	Type	Default	Description
(5)	Initial_stress			
	initial_stress_11	real	[0.0]	Component 11 (σ_{11})
	initial_stress_22	real	[0.0]	Component 22 (σ_{22})
	initial_stress_33	real	[0.0]	Component 33 (σ_{33})
	initial_stress_12	real	[0.0]	Component 12 (σ_{12})
	initial_stress_23	real	[0.0]	Component 23 (σ_{23})
	initial_stress_31	real	[0.0]	Component 31 (σ_{31})

• *List Read Method*

Material data must follow in the form:

< m, E(m), Pois(m), ρ(m), λ^w(m), ρ_w(m), n^w(m), Pf(m) >
< Dampm (m), Dampk (m) >
< (Stres(i,m), i = 1, 6) >
< terminate with a blank record >.

Notes/

(1) Poisson's ratio cannot be set equal to 1/2 since it results in division by zero. A value close to 1/2, say .4999, can be employed for incompressible applications.

(2) The element damping matrix is computed as:

$$\boldsymbol{c} = \text{Rayleigh\_mass\_damp}*\boldsymbol{m} + \text{Rayleigh\_stiffness\_damp}*\boldsymbol{k}$$

(3) Only applicable to porous media models.

(4) Isotropic viscoelasticity is simulated using an exponential (Prony) series for the bulk and shear functions. The viscoelastic data consists of the relaxation bulk and shear moduli values at specific relaxation times for each term in the series.

(5) Tensile stresses are positive

Notes . .

Notes . .

10.2 Stress_Model: Linear Orthotropic Elasticity Model

ORTHOTROPIC_ELASTIC

Material_name = ORTHOTROPIC_ELASTIC
Material_set_number = mset , etc...

Note	Variable Name	Type	Default	Description
• *Keywords Read Method*				
	Material_set_number	integer	[1]	Material set number ≤ Numat
	Mass_density	real	[0.0]	Mass density ρ
	Youngs_modulus	real	[0.0]	Young's modulus E
	Poissons_ratio	real	[0.0]	Poisson's ratio ν
(1)	Modulus_coefficient_C11	real	[0.0]	Modulus coefficient C11
	Modulus_coefficient_C22	real	[0.0]	Modulus coefficient C22
	Modulus_coefficient_C33	real	[0.0]	Modulus coefficient C33
	Modulus_coefficient_C44	real	[0.0]	Modulus coefficient C44
	Modulus_coefficient_C55	real	[0.0]	Modulus coefficient C55
	Modulus_coefficient_C66	real	[0.0]	Modulus coefficient C66
	Modulus_coefficient_C12	real	[0.0]	Modulus coefficient C12 (= C21)
	Modulus_coefficient_C23	real	[0.0]	Modulus coefficient C23 (= C32)
	Modulus_coefficient_C13	real	[0.0]	Modulus coefficient C13 (= C31)
(2)	Reference_direction_axes: n_x(1), n_y(1), n_z(1) n_x(2), n_y(2), n_z(2) n_x(3), n_y(3), n_z(3)	real	[ref.axes]	Material axes (if needed)
(3)	Mass_damping	real	[0.0]	Mass matrix Rayleigh damping coefficient
(3)	Stiffness_damping	real	[0.0]	Stiffness matrix Rayleigh damping coefficient
(4)	Solid_mass_density	real	[0.0]	Mass density (Solid Phase) ρ_s
(4)	Fluid_mass_density	real	[0.0]	Mass density (Fluid Phase) ρ_w
(4)	Fluid_bulk_modulus	real	[0.0]	Bulk modulus (Fluid Phase) λ^w
(4)	Porosity	real	[0.0]	Porosity n^w
(4)	Ref_fluid_pressure	real	[0.0]	Reference pore-fluid pressure
(4)	Pressure_load_time	integer	[0]	Pore-fluid pressure load time function

(cont'd)

(cont'd)

Note	Variable Name	Type	Default	Description
(5)	Initial_stress			
	initial_stress_11	real	[0.0]	Component 11 (σ_{11})
	initial_stress_22	real	[0.0]	Component 22 (σ_{22})
	initial_stress_33	real	[0.0]	Component 33 (σ_{33})
	initial_stress_12	real	[0.0]	Component 12 (σ_{12})
	initial_stress_23	real	[0.0]	Component 23 (σ_{23})
	initial_stress_31	real	[0.0]	Component 31 (σ_{31})

• *List Read Method*

Material data must follow in the form:

< m, E(m), Pois(m), ρ(m), λ^w(m), ρ_w(m), n^w(m), Pf(m) >
< C11(m), C22(m), …, C66(m) >
< C12(m), C23(m), C13(m) >
< i, n(1, m, i), n(2, m, i), n(3, m, i), i = 1 >
< i, n(1, m, i), n(2, m, i), n(3, m, i), i = 2 >
< i, n(1, m, i), n(2, m, i), n(3, m, i), i = 3 >
< Dampm (m), Dampk (m) >
< (Stres(i, m), i = 1, 6) >
< terminate with a blank record >.

Notes/
(1) If (C11*C22*...*C44 ≤ 0) the material defaults to an isotropic linear elastic model using Young's modulus and Poisson's ratio.

(2) Default is $\mathbf{n}_1 = \mathbf{e}_1 = \{1, 0, 0\}$, $\mathbf{n}_2 = \mathbf{e}_2 = \{0, 1, 0\}$, and $\mathbf{n}_3 = \mathbf{e}_3 = \{0, 0, 1\}$ where [$\mathbf{e}_1$, $\mathbf{e}_2$, $\mathbf{e}_3$] is the triad of unit base vectors used for the reference rectangular Cartesian axes. The orthotropic elasticity tensor **E**, is referred to the global coordinate axes via the rotation:

$$\mathbf{E'}_{ijkl} = \mathbf{E}_{klmn}\,\mathbf{R}_{ki}\,\mathbf{R}_{lj}\,\mathbf{R}_{mk}\,\mathbf{R}_{nl}$$

where $\mathbf{R} = [\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3]$. Note that the orthotropic direction vectors are restricted to be orthogonal to each other, viz.,

$$\mathbf{n}_I \cdot \mathbf{n}_J = \delta_{IJ}$$

(3) The element damping matrix is computed as:

$$\mathbf{c} = \text{Rayleigh\_mass\_damp}*\mathbf{m} + \text{Rayleigh\_stiffness\_damp}*\mathbf{k}$$

(4) Only applicable to porous media models.

(5) Tensile stresses are positive.

Notes . .

10.3 Stress_Model: Hyperelasticity Model

HYPERELASTIC

Material_name = HYPERELASTIC
Material_set_number = mset , etc...

The following hyperelastic stored energy function with uncoupled volumetric and deviatoric parts is employed:

$$\psi = 1/2\, G(I_b - 3) + U(J) \quad \text{with} \quad U(J) = B(JLnJ - J + 1)$$

where

$$J = \det(\boldsymbol{F}), \quad I_b = J^{-2/3}\boldsymbol{b} : \boldsymbol{g}, \quad \boldsymbol{b} = \boldsymbol{F}\boldsymbol{F}^t$$

It follows that the Kirchhoff stress is given by:

$$\tau = Jp\mathbf{g} + G\,J^{-2/3}\mathrm{dev}[\mathbf{b}] \quad \text{with} \quad p = dU(J)/dJ = BLnJ$$

Note	Variable Name	Type	Default	Description
• *Keywords Read Method*				
	Material_set_number	integer	[1]	Material set number ≤ Numat
	Hyperelastic_case	integer	[0]	Hyperelastic free energy function: = 0, $U(J) = B(JLnJ - J + 1)$ = 1, $U(J) = B(LnJ)^2 / 2$
	Mass_density	real	[0.0]	Mass density ρ
	Shear_modulus	real	[0.0]	Shear modulus G
	Bulk_modulus	real	[0.0]	Bulk modulus B
(1)	Solid_mass_density	real	[0.0]	Mass density (Solid Phase) ρ_s
(1)	Fluid_mass_density	real	[0.0]	Mass density (Fluid Phase) ρ_w
(1)	Fluid_bulk_modulus	real	[0.0]	Fluid bulk modulus λ^w
(1)	Porosity	real	[0.0]	Porosity n^w
(2)	Initial_stress			
	initial_stress_11	real	[0.0]	Component 11 (σ_{11})
	initial_stress_22	real	[0.0]	Component 22 (σ_{22})

(cont'd)

(cont'd)

Note	Variable Name	Type	Default	Description
	initial_stress_33	real	[0.0]	Component 33 (σ_{33})
	initial_stress_33	real	[0.0]	Component 33 (σ_{33})
	initial_stress_12	real	[0.0]	Component 12 (σ_{12})
	initial_stress_12	real	[0.0]	Component 12 (σ_{12})
	initial_stress_23	real	[0.0]	Component 23 (σ_{23})
	initial_stress_31	real	[0.0]	Component 31 (σ_{31})

• *List Read Method*

Material data must follow in the form:

< m, IHyper(m), G(m), B(m), ρ(m), λ^w(m), ρ_w(m), n^w(m), Pf(m) >
< (Stres(i, m), i = 1, 6) >
< terminate with a blank record >.

Notes/

(1) Only applicable to porous media models.

(2) Tensile stresses are positive.

10.4 Stress_Model: Mises Viscoelastic Creep Model

MCREEP

Material_name = MCREEP
Material_set_number = mset , etc...

The constitutive equation is written as follows:

$$\sigma = \mathbf{E} : (\varepsilon - \varepsilon^{c})$$

where **E** = isotropic elasticity tensor, ε = strain and ε^{c} = creep strain. The creep strain rate is given by:

$$\frac{d\boldsymbol{\varepsilon}^{c}}{dt} = \sqrt{\frac{3}{2}}\ \frac{d\bar{\varepsilon}^{c}}{dt}\ \frac{\boldsymbol{\sigma}}{|\boldsymbol{\sigma}|}$$

where $d\bar{\varepsilon}^{c}/dt$ = effective or equivalent creep strain rate. In the model it is assumed that the bulk response is linear elastic and that the nonlinear creep response is restricted to the deviatoric part of the stress and strain. The effective creep strain is assumed to consist of contributions from primary (transient) creep $\bar{\varepsilon}_{p}$, and secondary (steady-state) creep $\bar{\varepsilon}_{s}$, as follows:

$$\frac{d\bar{\varepsilon}^{c}}{dt} = \frac{d\bar{\varepsilon}_{p}}{dt} + \frac{d\bar{\varepsilon}_{s}}{dt}$$

with:

$$\frac{d\bar{\varepsilon}_{p}}{dt} = \left(A - B\bar{\varepsilon}_{p}\right)\frac{d\bar{\varepsilon}_{s}}{dt} \qquad \text{if } \frac{d\bar{\varepsilon}_{s}}{dt} \geq \frac{d\bar{\varepsilon}_{s}^{*}}{dt}$$

$$\frac{d\bar{\varepsilon}_{p}}{dt} = \left(A - B\,\frac{d\bar{\varepsilon}_{s}^{*}}{d\varepsilon_{s}}\,\bar{\varepsilon}_{p}\right)\frac{d\bar{\varepsilon}_{s}}{dt} \qquad \text{if } \frac{d\bar{\varepsilon}_{s}}{dt} < \frac{d\bar{\varepsilon}_{s}^{*}}{dt}$$

and

$$\frac{d\bar{\varepsilon}_{s}}{dt} = D\,\bar{\sigma}^{n}\,\exp\left(-\frac{Q}{RT}\right) \qquad \bar{\sigma} = \sqrt{\frac{3}{2}}\,|\text{dev}\,\boldsymbol{\sigma}|$$

where A, B, C, n, $d\bar{\varepsilon}_{s}^{*}/dt$ and Q are material parameters, T = absolute temperature (degrees Kelvin), and R = 1.987 cal/mole-K = universal gas constant. Values for the material parameters are given in Table 10.4.1.

Table 10.4.1

Material	Primary Creep			Secondary Creep		
	A	B	$d\bar{\varepsilon}_s^*/dt$ (sec.$^{-1}$)	D (Pa$^{-4.9}$/sec.)	n	Q (cal/mole)
Halite	4.56	127.	5.39E-8	5.79E-36	4.9	12.0E+3
Argillaceous Halite	4.56	127.	5.39E-8	1.74E-35	4.9	12.0E+3

The following hyperelastic stored energy function with uncoupled volumetric and deviatoric parts is employed:

$$\psi = 1/2\,\mathrm{G}\left(I_b^e - 3\right) + U\left(J^e\right) \quad \text{with} \quad U\left(J^e\right) = B\left(J^e LnJ^e - J^e + 1\right)$$

where

$$J^e = \det\left(\boldsymbol{F}^e\right), \quad I_b^e = J^{-2/3}\boldsymbol{b}^e : \boldsymbol{g}, \quad \boldsymbol{b}^e = \boldsymbol{F}^e\boldsymbol{F}^{et}$$

Note	Variable Name	Type	Default	Description
• *Keywords Read Method*				
	Material_set_number	integer	[1]	Material set number ≤ Numat
(1)	Hyperelastic_case	integer	[0]	Hyperelastic free energy function: = 0, $U(J) = B(JLnJ - J + 1)$ = 1, $U(J) = B(LnJ)^2 / 2$
	Mass_density	real	[0.0]	Mass density ρ
	Shear_modulus	real	[0.0]	Shear modulus G
	Bulk_modulus	real	[0.0]	Bulk modulus B
	Coefficient_A	real	[4.56]	Material parameter A
	Coefficient_B	real	[127.]	Material parameter B
	Coefficient_E	real	[5.39E-8]	Material parameter $d\bar{\varepsilon}_s^*/dt$
	Coefficient_D	real	[5.79E-36]	Material parameter D
	Coefficient_n	real	[4.9]	Power exponent n
	Coefficient_Q	real	[12.E+3]	Material parameter Q

(cont'd)

(cont'd)

Note	Variable Name	Type	Default	Description
	Reference_temperature	real	[273.15]	Reference Temperature T_0
(2)	Initial_stress			
	initial_stress_11	real	[0.0]	Component 11 (σ_{11})
	initial_stress_22	real	[0.0]	Component 22 (σ_{22})
	initial_stress_33	real	[0.0]	Component 33 (σ_{33})
	initial_stress_12	real	[0.0]	Component 12 (σ_{12})
	initial_stress_23	real	[0.0]	Component 23 (σ_{23})
	initial_stress_31	real	[0.0]	Component 31 (σ_{31})
(3)	Solid_mass_density	real	[0.0]	Mass density (Solid Phase) ρ_s
(3)	Fluid_mass_density	real	[0.0]	Mass density (Fluid Phase) ρ_w
(3)	Fluid_bulk_modulus	real	[0.0]	Fluid bulk modulus λ^w
(3)	Porosity	real	[0.0]	Porosity n^w
(3)	Ref_fluid_pressure	real	[0.0]	Reference pore-fluid pressure
(3)	Pressure_load_time	integer	[0]	Pore-fluid pressure load time function

• *List Read Method*

Material data must follow in the form:

< m, IHyper(m), G(m), B(m), ρ(m), λ^w(m), ρ_w(m), n^w(m), Pf(m) >
< A(m), B(m), E(m), D(m), n(m), Q(m), T_0(m) >
< (Stres(i, m), i = 1, 6) >
< terminate with a blank record >.

Notes/
(1) Only applicable to finite deformation case (see Section 9.2.1).

(2) Tensile stresses are positive.

(3) Only applicable to porous media models.

Notes . .

10.5 Stress_Model: Von Mises Elasto-(Visco-)Plastic Model

MISES

Material_name = MISES
Material_set_number = mset , etc...

The yield function is of the following type:

$$f = \bar{\tau} - c = 0$$

with

$$\bar{\tau} = \sqrt{J_2} \qquad J_2 = tr\left(s^2\right)/2 \qquad s = \tau - tr(\tau)/3\delta$$

where c (= cohesion) is a material constant, and τ = effective (Kirchhoff) stress. The following hyperelastic stored energy function with uncoupled volumetric and deviatoric parts is employed:

$$\psi = 1/2\,\mathrm{G}\left(I_b^e - 3\right) + U\left(J^e\right) \quad \text{with} \quad U\left(J^e\right) = B\left(J^e LnJ^e - J^e + 1\right)$$

where

$$J^e = \det\left(\boldsymbol{F}^e\right), \quad I_b^e = J^{-2/3}\boldsymbol{b}^e : \boldsymbol{g}, \quad \boldsymbol{b}^e = \boldsymbol{F}^e\boldsymbol{F}^{et}$$

Note	Variable Name	Type	Default	Description
• *Keywords Read Method*				
	Material_set_number	integer	[1]	Material set number ≤ Numat
(1)	Hyperelastic_case	integer	[0]	Hyperelastic free energy function: = 0, $U(J) = B(JLnJ - J + 1)$ = 1, $U(J) = B(LnJ)^2 \ / \ 2$
	Mass_density	real	[0.0]	Mass density ρ
	Shear_modulus	real	[0.0]	Shear modulus G
	Bulk_modulus	real	[0.0]	Bulk modulus B
	Activation_time	real	[0.0]	Time at which nonlinearities are activated.
	Cohesion	real	[0.0]	Cohesive coefficient *c*
	Shear_strength	real	[0.0]	Shear strength = *c*
	Axial_strength	real	[0.0]	Axial strength = $c\sqrt{3}$
	Relaxation_time	real	[0.0]	Relaxation time constant η > 0.0 ≤ 0.0, Elastoplastic > 0.0, Elastoviscoplastic
	Variable_cohesion	integer	[0]	Variable cohesion load time function number

---

(cont'd)

(cont'd)

Note	Variable Name	Type	Default	Description
(2)	Initial_stress:			
	initial_stress_11	real	[0.0]	Component 11 (σ_{11})
	initial_stress_22	real	[0.0]	Component 22 (σ_{22})
	initial_stress_33	real	[0.0]	Component 33 (σ_{33})
	initial_stress_12	real	[0.0]	Component 12 (σ_{12})
	initial_stress_23	real	[0.0]	Component 23 (σ_{23})
	initial_stress_31	real	[0.0]	Component 31 (σ_{31})
(3)	Solid_mass_density	real	[0.0]	Mass density (Solid Phase) ρ_s
(3)	Fluid_mass_density	real	[0.0]	Mass density (Fluid Phase) ρ_w
(3)	Fluid_bulk_modulus	real	[0.0]	Fluid bulk modulus λ^w
(3)	Porosity	real	[0.0]	Porosity n^w

• *List Read Method*

Material data must follow in the form:

< m, IHyper(m), G(m), B(m), ρ(m), λ^w(m), ρ_w(m), n^w(m), Pf(m) >
< c(m), η(m), ltime_coh(m) >
< (Stres(i, m), i = 1, 6) >
< terminate with a blank record >.

Notes/

(1) Only applicable to finite deformation case (see Section 9.2.1).

(2) Tensile stresses are positive.

(3) Only applicable to porous media models.

Notes . .

Notes . .

10.6 Stress_Model: Drucker-Prager Elasto-(Visco-)Plastic Model

DRUCKER_PRAGER

Material_name = DRUCKER_PRAGER
Material_set_number = mset , etc...

The yield function is of the following type:

$$f = \alpha p + \bar{\tau} - c = 0$$

where

$$\bar{\tau} = \sqrt{J_2} \qquad p = tr(\tau)/3 \qquad J_2 = tr\left(s^2\right)/2 \qquad s = \tau - p\delta$$

□ and c are material constants, τ = effective (Kirchhoff) stress. The Drucker-Prager cone which is internally tangential to the Mohr-Coulomb yield surface is obtained by setting:

$$\alpha = \sqrt{3}\sin\varphi / \sqrt{3 + \sin\varphi^2} \qquad c = C_{MC}\alpha / \tan\varphi$$

where C_{MC} = Mohr-Coulomb cohesion, φ = Mohr-Coulomb friction angle. The external cone is obtained by setting:

$$\alpha = 2\sqrt{3}\sin\varphi / (3 - \sin\varphi) \qquad c = C_{MC}\alpha / \tan\varphi$$

If $\beta = \alpha$ an associative flow rule is used. Otherwise a non-associative flow rule is used. The following hyperelastic stored energy function with uncoupled volumetric and deviatoric parts is employed:

$$\psi = 1/2\,\mathrm{G}\left(I_b^e - 3\right) + U\left(J^e\right) \quad \text{with} \quad U\left(J^e\right) = B\left(J^e LnJ^e - J^e + 1\right)$$

where

$$J^e = \det\left(\boldsymbol{F}^e\right), \quad I_b^e = J^{-2/3}\boldsymbol{b}^e : \boldsymbol{g}, \quad \boldsymbol{b}^e = \boldsymbol{F}^e\boldsymbol{F}^{et}$$

Note	Variable Name	Type	Default	Description
• *Keywords Read Method*				
	Material_set_number	integer	[1]	Material set number ≤ Numat
(1)	Hyperelastic_case	integer	[0]	Hyperelastic free energy function: = 0, $U(J) = B(JLnJ - J + 1)$ = 1, $U(J) = B(LnJ)^2 \ / \ 2$
	Mass_density	real	[0.0]	Mass density ρ
	Shear_modulus	real	[0.0]	Shear modulus G
	Bulk_modulus	real	[0.0]	Bulk modulus B
	Activation_time	real	[0.0]	Time at which nonlinearities are activated.
	Friction_angle	real	[0.0]	Friction angle $\varphi > 0.0$
	Cohesion	real	[0.0]	Cohesive coefficient *c*
	Dilation_angle	real	[0.0]	Dilation angle $\psi \geq 0.0$
	Internal_cone **on** / off	list	[on]	Internal cone option
	Tension_cutoff on / **off**	list	[off]	Tension cutoff options
	Relaxation_time	real	[0.0]	Relaxation time constant $\eta \geq 0.0$ ≤ 0.0, Elastoplastic > 0.0, Elastoviscoplastic
	Variable_cohesion	integer	[0]	Variable cohesion load time function number
	Variable_friction	integer	[0]	Variable friction angle load time function number
(2)	Initial_stress			
	initial_stress_11	real	[0.0]	Component 11 (σ_{11})
	initial_stress_22	real	[0.0]	Component 22 (σ_{22})
	initial_stress_33	real	[0.0]	Component 33 (σ_{33})
	initial_stress_12	real	[0.0]	Component 12 (σ_{12})
	initial_stress_23	real	[0.0]	Component 23 (σ_{23})
	initial_stress_31	real	[0.0]	Component 31 (σ_{31})

(cont'd)

(cont'd)

Note	Variable Name	Type	Default	Description
(3)	Solid_mass_density	real	[0.0]	Mass density (solid phase) ρ_s
(3)	Fluid_mass_density	real	[0.0]	Mass density (fluid phase) ρ_w
(3)	Fluid_bulk_modulus	real	[0.0]	Fluid bulk modulus λ^w
(3)	Porosity	real	[0.0]	Porosity n^w

• *List Read Method*

Material data must follow in the form:

< m, IHyper(m), G(m), B(m), ρ(m), λ^w(m), ρ_w(m), n^w(m), Pf(m) >
< ϕ(m), c(m), ψ(m), η(m), ltime_coh(m), ltime_phi(m) >
< (Stres(i, m), i = 1, 6) >
< terminate with a blank record >.

Notes/

(1) Only applicable to finite deformation case (see Section 9.2.1).

(2) Tensile stresses are positive.

(3) Only applicable to porous media models.

Notes . .

10.7 Stress_Model: Matsuoka Elasto-(Visco-)Plastic Model

MATSUOKA

Material_name = MATSUOKA
Material_set_number = mset , etc...

The yield function is of the following type :

$$f = c_y J_3 - (c_y - 3)(p - a_t)J_2 + (c_y - 9)(p - a_t)^3 = 0$$

where

$$\mathbf{s} = \tau - p\,\delta \qquad p = \mathrm{tr}(\tau)/3 \qquad J_2 = \mathrm{tr}(\mathbf{s}^2)/2 \qquad J_3 = \mathrm{tr}(\mathbf{s}^3)/3 = \det(\mathbf{s})$$

c_y and a_t are material constants. The Matsuoka cone which is closest to the Mohr-Coulomb yield surface is obtained by setting:

$$c_y = (9 - \sin\varphi^2) / (1 - \sin\varphi^2) \qquad a_t = c\ \tan\varphi$$

where c = Mohr-Coulomb cohesion, φ= Mohr-Coulomb friction angle. If the dilation angle $\psi = \phi$ □ an associative flow rule is used. Otherwise a non-associative flow rule is used. The following hyperelastic stored energy function with uncoupled volumetric and deviatoric parts is employed:

$$\psi = 1/2\,\mathrm{G}\left(I_b^e - 3\right) + U\left(J^e\right) \quad \text{with} \quad U\left(J^e\right) = B\left(J^e Ln J^e - J^e + 1\right)$$

where

$$J^e = \det\left(\boldsymbol{F}^e\right), \quad I_b^e = J^{-2/3}\boldsymbol{b}^e : \boldsymbol{g}, \quad \boldsymbol{b}^e = \boldsymbol{F}^e \boldsymbol{F}^{et}$$

References / Bibliography

1. Matsuoka, H. and T. Nakai, "Relationship Among Tresca, Mises, Mohr-Coulomb and Matsuoka-Nakai Failure Criteria," *Soils and Foundations*, **5**, No. 4, (1985), 123–128.

Note	Variable Name	Type	Default	Description
• *Keywords Read Method*				
	Material_set_number	integer	[1]	Material set number ≤ Numat
(1)	Hyperelastic_case	integer	[0]	Hyperelastic free energy function: $= 0,\ U(J) = B(JLnJ - J + 1)$ $= 1,\ U(J) = B(LnJ)^2 \ / \ 2$
	Tension_cutoff on / **off**	list	[off]	Tension cutoff options:
	Mass_density	real	[0.0]	Mass density ρ
	Shear_modulus	real	[0.0]	Shear modulus G
	Bulk_modulus	real	[0.0]	Bulk modulus B
	Activation_time	real	[0.0]	Time at which nonlinearities are activated.
	Friction_angle	real	[0.0]	Friction angle $\varphi > 0.0$
	Cohesion	real	[0.0]	Cohesive coefficient c
	Dilation_angle	real	[0.0]	Dilation angle $\psi \geq 0.0$
	Relaxation_time	real	[0.0]	Relaxation time constant $\eta \geq 0.0$ ≤ 0.0, Elastoplastic > 0.0, Elastoviscoplastic
	Variable_cohesion	integer	[0]	Variable cohesion load time function number
	Variable_friction	integer	[0]	Variable friction angle load time function number
(2)	Initial_stress			
	initial_stress_11	real	[0.0]	Component 11 (σ_{11})
	initial_stress_22	real	[0.0]	Component 22 (σ_{22})
	initial_stress_33	real	[0.0]	Component 33 (σ_{33})
	initial_stress_12	real	[0.0]	Component 12 (σ_{12})
	initial_stress_23	real	[0.0]	Component 23 (σ_{23})
	initial_stress_31	real	[0.0]	Component 31 (σ_{31})
(3)	Solid_mass_density	real	[0.0]	Mass density (Solid Phase) ρ_s
(3)	Fluid_mass_density	real	[0.0]	Mass density (Fluid Phase) ρ_w
(3)	Fluid_bulk_modulus	real	[0.0]	Fluid bulk modulus λ^w
(3)	Porosity	real	[0.0]	Porosity n^w

(cont'd)

(cont'd)

Note	Variable Name	Type	Default	Description

• *List Read Method*

Material data must follow in the form:

< m, IHyper(m), G(m), B(m), ρ(m), λ^w(m), ρ_w(m), n^w(m), Pf(m) >
< ϕ(m), c(m), ψ(m), η(m). ltime_coh(m), ltime_phi(m) >
< (Stres(i, m), i = 1, 6) >
< terminate with a blank record >.

Notes/

(1) Only applicable to finite deformation case (see Section 9.2.1).

(2) Tensile stresses are positive.

(3) Only applicable to porous media models.

Notes . .

10.8 Stress_Model: Cap Model (Geomaterials)

Following common usage in geomechanics, compressive stresses and strains are taken as positive in the following. The following hyperelastic stored energy function is employed:

$$\psi\left(\varepsilon_v^e,\varepsilon_s^e\right)= p_1\frac{\kappa}{\beta_1}\exp\left(\frac{\varepsilon_v^e}{\kappa}\right)+\frac{3}{2}\mu\left(\varepsilon_s^e+\varepsilon_{s1}^e\right)^2 \tag{1}$$

where

$$\varepsilon_v^e = tr\varepsilon^e;\ \varepsilon_s^e=\sqrt{\frac{2}{3}tr\left(\boldsymbol{e}^e\right)^2};\ \boldsymbol{e}^e=\varepsilon^e-\frac{1}{3}tr\varepsilon^e\delta \tag{2}$$

κ = elastic compressibility index, and $\mu=\mu\left(\varepsilon_v^e\right)=$ elastic shear modulus defined by the expression:

$$\mu=\mu_0+p_1\frac{\alpha}{\beta_1}\exp\left(\frac{\varepsilon_v^e}{\kappa}\right) \tag{3}$$

where $\alpha=$ material parameter which describes the variation of the shear modulus with the elastic volumetric strain. In Eq. 1, ε_{s1}^e and β_1 are parameters corresponding to the strain-free reference state $\left(p_1,q_1\right)$ at which:

$$\varepsilon_{s1}^e=\frac{q_1}{3\mu_1};\ \beta_1=1+\frac{3}{2}\frac{\alpha}{\kappa}\left(\varepsilon_{s1}^e\right)^2;\ \mu_1=\mu_0+\frac{\alpha}{\beta_1}p_1 \tag{4}$$

and

$$p=\frac{1}{3}tr\sigma=\text{mean stress};\ q=\sqrt{3J_2}=\text{shear stress} \tag{5}$$

with

$$\sigma=\text{effective stress};\ \boldsymbol{s}=\sigma-p\delta=\text{deviatoric stress};\ J_2=tr\left(s^2\right)/2 \tag{6}$$

The elasto-plasticity is defined in terms of a yield function and a failure criterion as illustrated on Fig. 10.8.1.

1. Yield Function The yield function is of the following form:

$$f_{cap} = f_{cap}(p,q,p_c) = q^2 - M^2(p+a)(p_c - p) \tag{7}$$

where M = material parameter (= slope of critical state line), and p_c = preconsolidation stress = scalar plastic state variable describing the size of the yield function. An associative flow rule is used, and the following hardening rule is employed:

$$\frac{\dot{p}_c}{p_c} = \frac{\dot{\varepsilon}_v^p}{\lambda - \kappa} \tag{8}$$

where $\dot{\varepsilon}_v^p$ = plastic volumetric strain rate; λ = compressibility index for virgin loading. The cap yield function plots in stress space as an ellipsoid as in the modified Cam_clay theory (Roscoe and Burland, 1968).

2. Failure Criterion The failure criterion is of the following form:

$$f_{fail} = f_{fail}(p,q) = q - M(p+a) \tag{9}$$

and a non-associative flow rule is used. The failure criterion plots in stress space as a circular cone of the Drucker-Prager (1952) type.

In Eqs. 7 and 9, a = attraction = c/tan φ where c = cohesion, and φ = friction angle. The slope M can be selected to fit the corresponding Mohr-Coulomb criterion by setting:

$$M = \frac{6\sin\varphi}{3 - \sin\varphi} \qquad : \text{ external cone} \tag{10a}$$

$$M = \frac{3\sin\varphi}{\left(3 + \sin^2\varphi\right)^{\frac{1}{2}}} \qquad : \text{ internal cone} \tag{10b}$$

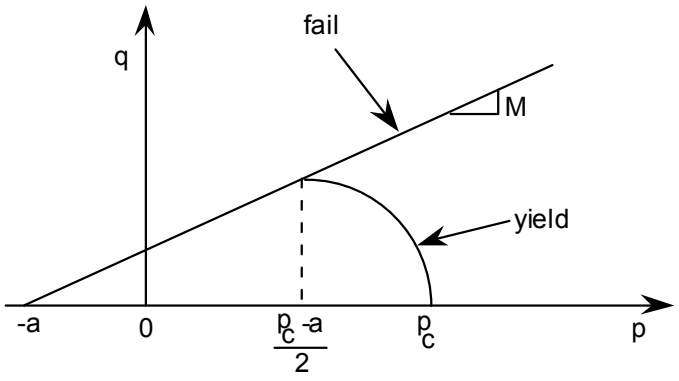


Figure 10.8.1 Cap Model

Notes/
(1) The compressibility indices λ and κ are related to the slopes of the virgin loading and unloading-reloading curves on the $[e \text{ vs } \log_{10} p]$ plot as measured in one-dimensional or hydrostatic consolidation tests as:

$$\lambda = \frac{C_c}{1+e_0}\frac{1}{Ln10} \quad \kappa = \frac{C_s}{1+e_0}\frac{1}{Ln10} \tag{11}$$

where

$$C_i = -\frac{\partial e}{\partial(\log_{10} p)}; \quad e = \text{void ratio} \tag{12}$$

$C_i = C_c \text{ and } C_s$ for virgin compression and swelling, respectively; $e_0 =$ void ratio at the reference mean stress p_0.

(2) The following variations of the elastic bulk and shear moduli with the mean stress p can be derived from Eqs. 1 and 3:

$$B = \frac{1}{\kappa}p; \; \mu = \mu_0 + \frac{\alpha}{\beta}p \tag{13}$$

where

$$\beta = 1 + \frac{3}{2}\frac{\alpha}{\kappa}\left(\varepsilon_s^c + \varepsilon_{s1}^c\right)^2 \tag{14}$$

Thus, with a suitable choice of parameters, the elastic shear modulus μ can be made either a constant or a linear function of p.

References / Bibliography

1. Borja, R.I., C. Tamagnini, and A. Amorosi, "Coupling Plasticiy and Energy-Conserving Elasticity Models for Clays", *J. Geotech. Geoenvironmental Eng.*, ASCE, Vol. 123, No. 10, (1997), pp. 948–957.

2. Drucker, D.C. and V. Prager, "Soil Mechanics and Plastic Analysis or Limit Design", *Quarterly of Applied Mathematics*, Vol. 10, (1952), pp. 157–165.

3. Roscoe, K.H. and J.H. Burland, "On the Generalized Stress-Strain Behavior of Wet Clays," *Engineering Plastaity,* J. Heyman and F.A. Leckie, Eds., Cambridge University Press, London, England, (1968), pp. 535–609.

4. Scofield, A. and P. Wroth, *Critical State Soil Mechanics,* McGraw-Hill, Inc., New York, NY, (1968).

CAP

Material_name = CAP
Material_set_number = mset , etc...

The following data are used to describe the Cap model.

Note	Variable Name	Type	Default	Description
• *Keywords Read Method*				
	Material_set_number	integer	[1]	Material set number ≤ Numat
(1)	Hyperelastic_case	integer	[0]	Hyperelastic free energy function
	Solid_mass_density	real	[0.0]	Solid mass density
	Shear_modulus	real	[0.0]	Shear modulus μ
	Bulk_modulus	real	[0.0]	Bulk modulus B
	Friction_angle	real	[0.0]	Friction angle $\varphi > 0.0$
	Cohesion	real	[0.0]	Cohesion c ≥ 0.0
	Dilation_angle	real	[0.0]	Dilation angle $\psi \geq 0.0$
	Internal_cone **on** / off	list	[on]	Internal cone option
	Tension_cutoff on / **off**	list	[off]	Tension cutoff option

(cont'd)

Notes/
(1) Only applicable to finite deformation case (see Section 9.2.1).

(cont'd)

Note	Variable Name	Type	Default	Description
(2)	Porosity	real	$[e_0 / (1+e_0)]$	Porosity n_0
(2)	Void_ratio	real	$[n_0 /(1-n_0)]$	Void ratio e_0
	Ref_mean_stress	real	[0.0]	Reference mean stress $p_0 > 0.0$
(3)	OCR	real	[1.0]	Overconsolidation ratio OCR ≥ 1
(4)	Compression_index	real	[0.0]	Compression index C_c
(4)	Swelling_index	real	[0.0]	Swelling index $C_s < C_c$
	Variable_shear_modulus	real	[0.0]	Variable shear modulus coefficient $\alpha \geq 0$

(cont'd)

Notes/

(2) Either the porosity or the void_ratio must be specified.

(3) The preconsolidation mean stress p_{c0} at the reference state p_0 is computed as:

$$p_{c0} = OCR\, p_0$$

(4) The compression index C_c and the swelling index C_s are defined as the slope of virgin loading and unloading-reloading curves, respectively, on the $[e \text{ vs } \log_{10} p]$ plot as measured in one-dimensional or hydrostatic consolidation tests, where $e =$ void ratio; $p =$ mean stress, viz.,

$$C_c = -\left.\frac{\partial e}{\partial(\log_{10} p)}\right|_{virgin} \qquad C_s = -\left.\frac{\partial e}{\partial(\log_{10} p)}\right|_{unload}$$

(cont'd)

Note	Variable Name	Type	Default	Description
(5)	Initial_stress			
	initial_stress_11	real	[0.0]	Component 11 (σ_{11})
	initial_stress_22	real	[0.0]	Component 22 (σ_{22})
	initial_stress_33	real	[0.0]	Component 33 (σ_{33})
	initial_stress_12	real	[0.0]	Component 12 (σ_{12})
	initial_stress_23	real	[0.0]	Component 23 (σ_{23})
	initial_stress_31	real	[0.0]	Component 31 (σ_{31})
(6)	Fluid_mass_density	real	[0.0]	Mass density (fluid phase) ρ_w
(6)	Fluid_bulk_modulus	real	[0.0]	Fluid bulk modulus λ^w

Notes/
(5) Tensile stresses are positive. If all are equal to zero, set internally equal to $-p_0$.

(6) Only applicable to porous media models.

EXAMPLE

```
Define_Material_Model  /
   name = "cam_clay"  /
   number_of_material_sets = 1

   stress_model  /
   material_name = cap  /
   material_type = nonlinear

      material_set_number = 1  /
      shear_modulus = 8.174e+02  /
      bulk_modulus = 2.180e+03  /
      friction_angle = 40.0  /
      dilation_angle = 30.0  /
      void_ratio = 0.889  /
      ref_mean_stress = 30.0  /
      OCR = 1  /
      variable_shear_modulus = 0.0  /
      compression_index = 0.40  /
      swelling_index = 0.06  /
      internal_cone = off
```

Notes . .

10.9 Stress_Model: Multi-Yield Elasto-Plastic Models (Geomaterials)

The following hyperelastic stored energy function with uncoupled volumetric and deviatoric parts is employed:

$$\psi = 1/2\,\mathrm{G}\left(I_b^e - 3\right) + U\left(J^e\right) \quad \text{with} \quad U\left(J^e\right) = B\left(J^e LnJ^e - J^e + 1\right)$$

where

$$J^e = \det\left(\boldsymbol{F}^e\right), \quad I_b^e = J^{-2/3}\boldsymbol{b}^e : \boldsymbol{g}, \quad \boldsymbol{b}^e = \boldsymbol{F}^e\boldsymbol{F}^{et}$$

In the small deformation case, the form of the material constitutive tangent moduli tensor $\boldsymbol{K}$ is given as follows:

$$\boldsymbol{K} = \boldsymbol{E} - (\boldsymbol{E} : \boldsymbol{P})(\boldsymbol{Q} : \boldsymbol{E})/(H' + \boldsymbol{Q} : \boldsymbol{E} : \boldsymbol{P})$$

in which H' = plastic modulus; $\boldsymbol{P}$ and $\boldsymbol{Q}$ = symmetric second-order tensors such that $\boldsymbol{P}$ gives the direction of plastic deformations, $\boldsymbol{Q}$ is the outer normal to the active yield surface; $\boldsymbol{E}$ = fourth-order tensor of elastic moduli, assumed isotropic for the particular class of material models implemented.

The plastic potential is selected such that:

$$\boldsymbol{P} - 1/3(tr\boldsymbol{P})\delta = \boldsymbol{Q} - 1/3(tr\boldsymbol{Q})\delta = \boldsymbol{Q}'$$

Several material models have been implemented and may be selected by specifying the value of the control parameter Plasticity_sub_type as follows:

(1) Plasticity_sub_type = 1 to 5: <u>pressure non-sensitive materials.</u>

The yield function in this case is of the Von Mises type with:

$$f = \left\{ 3/2 tr\left(\boldsymbol{s} - \alpha\right)^2 \right\}^{1/2} - \; k = 0$$

where $\boldsymbol{s}$ is the deviatoric stress tensor, i.e.,

$$\boldsymbol{s} = \sigma - p\delta \qquad p = 1/3 tr\sigma$$

α is the coordinate of the center of the yield surface in the deviatoric stress space; and k is the size of the yield surface.

(2) Plasticity_sub_type = 8: <u>pressure sensitive materials.</u>

The yield function in this case is of the Drucker-Prager / Mohr_Coulomb type with (see Ref. [5]):

$$\mathrm{f} = \left\{ 3/2\,\mathrm{tr}\left(\boldsymbol{s} - \bar{\mathrm{p}}\alpha\right)^2 \right\}^{1/2} + \; \mathrm{k}\bar{\mathrm{p}}\mathrm{g}(\theta) = 0$$

where $\bar{p} = (p - a)$, $a =$ attraction $= c / \tan\varphi$. The function $g(\theta)$ determines the shape of the cross-section on the deviatoric plane and

$$g(\theta) = \frac{2M_k}{(1+M_k) - (1-M_k)\sin 3\theta}$$

in which

$$\sin 3\theta = -\sqrt{6}\ \bar{J}_3 / \bar{J}_2^{3/2}$$

$$\bar{J}_2 = tr\bar{\boldsymbol{s}}^2 \qquad \bar{J}_3 = tr\bar{\boldsymbol{s}}^3 \qquad \bar{\boldsymbol{s}} = \boldsymbol{s} - \bar{\mathrm{p}}\alpha$$

and $M_k =$ material parameter. For a Drucker-Prager circular cone: $M_k = 1.0$, whereas for a round-cornered Mohr-coulomb cone:

$$M_k = \frac{3 - \sin\varphi}{3 + \sin\varphi}$$

where $\varphi =$ friction angle.

Several different plastic potential functions may be selected by specifying the value of the Plastic_potential_code as follows:

- Plasticity_potential_code = 0:
  The plastic potential in this case is selected as follows:

  $$tr\boldsymbol{P} = X_{pp}\left(\bar{\eta}^2 - 1\right) / \left(\bar{\eta}^2 + 1\right)$$

  in which $\bar{\eta} =$ normalized stress ratio, viz.,

  $$\bar{\eta} = \eta / \eta_\psi \qquad \eta = \left\{\frac{3}{2} tr\boldsymbol{s}^2\right\}^{1/2} / \bar{p}$$

  with $\eta_\psi =$ dilation stress ratio, and $X_{pp} = X_{pp}\left(\bar{\varepsilon}^p\right) =$ dilation parameter (see Ref. [6]) as:

  $$\mathrm{X_{pp}} = \mathrm{X_{ppo}}\ \exp\left(\mathrm{d}\,\bar{\varepsilon}^{\mathrm{p}}\right) \ \le \mathrm{X_{pp}}\_max$$

  in which $d =$ damage_rate, and $\bar{\varepsilon}^p =$ cumulative plastic shear distortions, viz.,

  $$\bar{\varepsilon}^p = \int \left\{\frac{2}{3} tr\left(\dot{\boldsymbol{e}}^p\right)^2\right\}^{1/2} dt$$

  with

  $$\dot{\boldsymbol{e}}^p = \dot{\varepsilon}^p - \frac{1}{3} tr\dot{\varepsilon}^p \dot{\varepsilon} = \text{ plastic shear distortion rate.}$$

- Plastic_potential_code = 1:
  The plastic potential in this case is selected as follows:

1. Compactive phase: $\bar{\eta} \le 1$

$$tr\boldsymbol{P} = -X_{pp}\sqrt{1-\bar{\eta}^2}$$

2. Dilative phase: $\bar{\eta} > 1$

$$tr\boldsymbol{P} = X_{pp}\sqrt{\bar{\eta}^2 - 1} \quad if \quad tr(\boldsymbol{s} \bullet \bar{\boldsymbol{s}}) \ge 0 \quad \text{(loading case)}$$

$$tr\boldsymbol{P} = tr\boldsymbol{Q} \quad if \quad tr(\boldsymbol{s} \bullet \bar{\boldsymbol{s}}) < 0 \quad \text{(unloading case)}$$

where $\boldsymbol{Q} = \partial_\sigma \mathrm{f} =$ outer normal to the yield surface.

As previously, $X_{pp} = X_{pp}(\bar{\varepsilon}^p) =$ dilatational parameter, which is scaled according to the level of confinement as follows:

$$X_{pp} = X_{pp} \exp[1 - p/p_1]$$

with p_1 = reference mean normal stress.

A collection of nested yield surfaces may be used. This allows for the adjustment of the plastic hardening rule to any experimental hardening data; for example, data obtained from axial or simple shear tests. It is assumed that the yield surfaces are all similar, and that a plastic modulus (H') is associated with each one.

Several different plastic hardening rules may be selected by specifying the value of Plasticity_sub_type, as indicated below:

- Plasticity_sub_type= 1: Isotropic hardening rule
  The yield surfaces in this case do not change position, but merely increase in size as loading proceeds.

- Plasticity_sub_type = 2: Isotropic hardening/softening rule
  This case is a generalization of the previous model in which softening starts to occur when the outermost yield surface is reached. At this point, the elasto-plastic shear modulus is set to be $H = \delta_1 G$, and remains constant until k .LE. δ_2. Thereafter $H = 0$.

- Plasticity_sub_type = 3: Kinematic hardening rule
  In this case, the yield surfaces do not change size, but are translated in stress space by the stress point.

- Plasticity_sub_type = 4: Kinematic hardening/Isotropic softening rule
  In this case a combination of kinematic/isotropic hardening laws is used. A nonlinear isotropic hardening/softening model is adopted in which a saturation hardening/softening function of the exponential type is used as follows:

$$k \equiv k(\bar{\varepsilon}^p) = k_o[r + (1-r)\exp(-\delta\bar{\varepsilon}^p)]$$

where r = k_∞ / k_0 = reduction strength ratio (r > 0), and □ = reduction_rate

$(\delta \geq 0)$ are material parameters; $\bar{\varepsilon}^p$ = cumulative plastic shear distortions.

- Plasticity_sub_type= 5: Kinematic hardening/Isotropic softening rule
  The particular material model implemented in that option assumes cyclic degradation of the material properties according to the rule:

$$\mathrm{d}\tau / \tau = -\delta_1 \left(\gamma^{\delta 2}\right) \mathrm{d}(\log \mathrm{N})$$

  as observed in cyclic strain-controlled simple shear soil tests; τ = Shear stress amplitude; γ = Shear strain amplitude; N = Number of cycles.

- Plasticity_sub_type = 8: Kinematic hardening rule
  A purely kinematic hardening is adopted for that model. The dependence of the moduli on the effective mean normal stress is assumed of the following form:

$$x = x_1 \left(p / p_1\right)^n$$

  where $x = G,\ B,\ H'$, and n = power exponent, a material constant (see e.g., Ref. [9]). The following may be used as estimates: for cohesionless soils $n \approx 1/2$, and $n \approx 1$ for cohesive soils.

References / Bibliography

1. Prevost, J.H.,"Plasticity Theory for Soil Stress-Strain Behavior," *J. Eng. Mech. Div.*, ASCE, Vol. 104, No. EM5, (1978), pp. 1177-1194.

2. Prevost, J.H., and T.J.R. Hughes,"Finite Element Solution of Elastic-Plastic Boundary Value Problems," *J. Appl. Mech.*, ASME, Vol. 48, No. 1, (1981), pp. 69-74.

3. Prevost, J.H.,"Nonlinear Transient Phenomena in Elastic-Plastic Solids," *J. Eng. Mech. Div.*, ASCE, Vol. 108, No. EM6, (1982), pp. 1297-1311.

4. Prevost, J.H.,"Localization of Deformations in Elastic-Plastic Solids," *Int. J. Num. Meth. Eng.*, Vol. 8, No. 2, (1984), pp. 187-196.

5. Prevost, J.H.,"A Simple Plasticity Theory for Frictional Cohesionless Soils," *Int. J. Soil Dyn. Earth. Eng.*, Vol. 4, No. 1, (1985), pp. 9-17.

6. Popescu, R., and J.H. Prevost, "Centrifuge Validation of a Numerical Model for Dynamic Soil Liquefaction," *Soil Dyn. and Earth. Eng.*, Vol. 12, (1993), pp. 73-90.

7. Prevost, J.H. and C.M. Keane, "Shear Stress-Strain Curve Generation from Simple Material Parameters," *J. Geotech. Eng.*, ASCE, Vol. 116, (1990), pp. 1255-1263.

8. Hayashi, H., M. Honda, T. Yamada, and F. Tatsuoka, "Modeling of Nonlinear Stress Strain Relations of Sands for Dynamic Response Analysis," *Proceedings, 10th WCEE,* Madrid, Spain, Vol. 11, (1994), pp. 6819-6825.

9. Richart, F.E., J.R. Hall and R.D. Woods, *Vibrations of Soils and Foundations*, Prentice-Hall, (1970).

MULTI_YIELD

Material_name = MULTI_YIELD Max_number_of_yield_surfaces = Nys_max
Material_set_number = mset , etc...

The maximum number of yield surfaces for all materials in the set must be provided following the material name.

Note	Variable Name	Type	Default	Description
• *Keywords Read Method*				
	Material_set_number	integer	[1]	Material set number ≤ Numat
(1)	Hyperelastic_case	integer	[0]	Hyperelastic free energy function: = 0, $U(J) = B(JLnJ - J + 1)$ = 1, $U(J) = B(LnJ)^2 \ / \ 2$
	Mass_density	real	[0.0]	Mass density ρ
(2)	Shear_modulus	real	[0.0]	Shear modulus G_1
(2)	Bulk_modulus	real	[0.0]	Bulk modulus B_1
	Activation_time	real	[0.0]	Time at which nonlinearities are activated.
(3)	Initial_stress			
	initial_stress_11	real	[0.0]	Component 11 (σ_{11})
	initial_stress_22	real	[0.0]	Component 22 (σ_{22})
	initial_stress_33	real	[0.0]	Component 33 (σ_{33})
	initial_stress_12	real	[0.0]	Component 12 (σ_{12})
	initial_stress_23	real	[0.0]	Component 23 (σ_{23})
	initial_stress_31	real	[0.0]	Component 31 (σ_{31})
(4)	Solid_mass_density	real	[0.0]	Mass density (Solid Phase) ρ_s
(4)	Fluid_mass_density	real	[0.0]	Mass density (Fluid Phase) ρ_w
(4)	Fluid_bulk_modulus	real	[0.0]	Fluid bulk modulus λ^w
(4)	Porosity	real	[0.0]	Porosity n^w
	Number_of_yield_surfaces	integer	[Nys_max]	Number of yield surfaces ≥ 0 and ≤ Nys_max
	Yield_type **Mises** Drucker_Prager Mohr_Coulomb	list	[*]	Yield function type

(cont'd)

(cont'd)

Note	Variable Name	Type	Default	Description
	Plasticity_sub_type	integer	[3]	Plasticity material sub-type ≥ 1 and ≤ 8
	Principal_anisotropy	integer	[2]	Principal cross-anisotropy direction.
• *Plasticity_sub_type = 4*				
	Reduction_ratio	real	[0.0]	Reduction strength ratio k_∞ / k_0
	Reduction_rate	real	[0.0]	Reduction rate $\delta \geq 0$
• *Plasticity_sub_type = 8*				
	Internal_cone **on** / off	list	[on]	Internal cone option (only applicable to Drucker-Prager yield function type)
	Plastic_potential_code	integer	[0]	Plastic potential code ≥ 0 and ≤ 1 = 0: standard; =1: enhanced.
(5)	Ref_mean_stress	real	[0.0]	Reference mean normal stress $p_1 > 0.0$
	Power_exponent	real	[0.0]	Power exponent $n > 0.0$
	Cohesion	real	[0.0]	Cohesive coefficient $c > 0.0$
	Friction_angle_comp	real	[0.0]	Ultimate friction angle in compression $\phi_c > 0.0$
	Friction_angle_ext	real	[ϕ_c]	Ultimate friction angle in extension $\phi_e > 0.0$
	Dilation_angle_comp	real	[0.0]	Dilation angle in compression $\psi_c \geq 0.0$
	Dilation_angle_ext	real	[0.0]	Dilation angle in extension $\psi_e \geq 0.0$
(6)	Dilatational_parameter_Xpp	real	[1.0]	Dilatational parameter $X_{pp} \geq 0.0$
	Max_dilatational_Xpp	real	[Xpp]	Maximum dilatational parameter $X_{pp\_max} \geq 0.0$
	Dilatational_ratio	real	[1.0]	Dilatational ratio
	Damage_rate	real	[0.0]	Damage rate d

(cont'd)

(cont'd)

Note	Variable Name	Type	Default	Description
• *Shear Stress-Strain Generation Data*				
	Number_of_generation_pts	integer	[100]	Number of generation points
	Stress_driven **on** / off	list	[on]	Stress / strain driven option
	Shear stress-strain generation data must follow			

• *List Read Method*
Material data must follow in the form:
< m, Nys(m), IHyper(m), G(m), B(m), ρ(m), λ^w(m), ρ_w(m), n^w(m), Pf(m), cpt(m) >
if (Plasticity_sub_type le 5) then
< τ_max(m), γ_max(m), α(m), xl(m), xu(m) >
if (Plasticity_sub_type eq 8) then
< c(m), p1(m), η(m), ψ_c(m), ψ_e(m), Xpp(m), Xpp_comp(m), Xpp_ext(m) >
< ϕ_c(m), ϕ_e(m), K_0(m), Slope(m), γmax_c(m), γmax_e(m),
α_c(m), x1_c(m), xu_c(m), α_e(m), x1_e(m), xu_e(m) >
< Stres(i, m), i = 1, 6) >
< terminate with a blank record >.				

Notes/

(1) Only applicable to finite deformation case (see Section 9.2.1).

(2) For Plasticity_sub_type ≥ 6, G_1 and B_1 are the elastic shear and bulk moduli at the reference mean stress p_1 (see Note 5).

(3) Tensile stresses are positive.

(4) Only applicable to porous media models.

(5) The dependence of the elastic shear and bulk moduli on the (effective) mean normal stress is assumed of the following form:

$$G = G_1(p/p_1)^n \qquad B = B_1(p/p_1)^n$$

(6) See Ref. [6] for details.

Shear Stress-Strain Data Generation:

For the shear stress-strain curve generation, given G_1 = maximum shear modulus, τ_{max} = maximum shear stress, and γ_{max} = maximum shear strain, two options are available as follows:

Option 1: Let $y = \tau / (G_1 \gamma_{max})$ and $x = \gamma / \gamma_{max}$, then

$$y = y_i \frac{x}{y_1 + x} - \frac{y_1^2}{(y_1 + 1)^2} \frac{x^{m+1}}{m+1}$$

where the parameter y_1 is determined by requiring that at x =1, y = y_{max} as detailed in Ref. [7].

Option 2: Let $y = \tau / \tau_{max}$ and $x = \gamma / \gamma_r$ with $\gamma_r = \tau_{max} / G_1$, then:

$$y = e^{-\alpha x} f(x, x_1) + \left(1 - e^{-\alpha x}\right) f(x, x_u)$$

with

$$f(x, x_i) = \left[(2x/x_i + 1)^{x_i} - 1\right] / \left[(2x/x_i + 1)^{x_i} + 1\right]$$

where x_i, x_1, and x_u are material parameters as detailed in Ref [8].

Note	Variable Name	Type	Default	Description
• *Plasticity_sub_type = 1 to 5*				
	Max_shear_stress	real	[0.0]	Maximum shear stress $\tau_{max} > 0.0$
	Max_shear_strain	real	[0.06]	Maximum shear strain $\gamma_{max} > (\tau_{max}/G_1)$
(7)	Coefficient_alpha	real	[0.0]	Generation coefficient $\alpha \geq 0.0$
	Coefficient_x1	real	[0.30]	Generation coefficient $x_1 \geq 0.0$
	Coefficient_xu	real	[1.0]	Generation coefficient $x_u \geq 0.0$

Notes/
(7) If α = 0.0, the generation option 1 is used by default.

For Plasticity_sub_type = 8, the shear stress-strain data are generated at the reference mean normal stress p_1. The maximum shear stress at the reference (effective) mean normal stress p_1 is computed as follows: Let: σ_v = (effective) vertical stress; σ_h = (effective) horizontal stress; and following common usage in geotechnical engineering, assume that compressive stresses are counted as positive. Then the mobilized friction angle ϕ is computed as:

$$\sin\phi = \left(\sigma_v - \sigma_h\right) \Big/ \left(2a + \left(\sigma_v + \sigma_h\right)\right); \qquad a = c / \tan\phi_c = \text{attraction}$$

Note that in the above expression $\phi > 0$ is positive in compressioon, and $\phi < 0$ is negative in extension. Let:

$$p = \left(\sigma_v + 2\sigma_h\right)/3 = \text{mean stress}; \qquad q = \left(\sigma_v - \sigma_h\right) = \text{shear stress}$$

Initially

$$p_1 = \sigma_v (1 + 2K_0) / 3 ; \qquad q_1 = \sigma_v (1 - K_0) = 3p_1(1 - K_0) / (1 + 2K_0)$$

and at failure (ultimate state), the maximum shear stress (τ_{max}) is computed as $\tau_{max} = |q_{max}|$, with:

$$q_{max} = 2\sin\phi_{max} (a + a_1) / (1 - \sin\phi_{max} (2S + 1/3)); \qquad a_1 = p_1 - Sq_1$$

where S = slope of axial stress path followed in the test (see Note 9); $\phi_{max} = \phi_c$ in compression tests and $\phi_{max} = -\phi_e$ in extension tests, respectively.

Note	Variable Name	Type	Default	Description
• *Plasticity_sub_type = 8*				
(8)	Lateral_stress_coefficient	real	[ν/(1-ν)]	Coefficient of lateral stress $K_0 \geq 0.0$
(9)	Axial_stress_path_slope	real	[0.0]	Slope of axial stress path ≥ 0.0
	Max_shear_strain_comp	real	[0.06]	Max shear strain in compression $\gamma_{maxc} \geq 0.0$
	Max_shear_strain_ext	real	[γ_{maxc}]	Max shear strain in extension $\gamma_{maxe} \geq 0.0$
(10)	Coefficient_alpha_comp	real	[0.0]	Generation coefficient in compression $\alpha_c \geq 0.0$
	Coefficient_x1_comp	real	[0.30]	Generation coefficient in compression $x_{1c} \geq 0.0$
	Coefficient_xu_comp	real	[1.00]	Generation coefficient in compression $x_{uc} \geq 0.0$
	Coefficient_alpha_ext	real	[α_c]	Generation coefficient in extension $\alpha_e \geq 0.0$
	Coefficient_x1_ext	real	[0.30]	Generation coefficient in extension $x_{1e} \geq 0.0$
	Coefficient_xu_ext	real	[1.0]	Generation coefficient in extension $x_{ue} \geq 0.0$

Notes/

(8) If $K_0 = 0.0$, set internally equal to elastic $K_0 = \nu / (1 - \nu)$, with ν = Poisson's ratio:

$$\nu = (3B_1 - 2G_1) / 2(3B_1 + G_1) ; \qquad K_0 = (3B_1 - 2G_1) / (3B_1 + 4G_1)$$

(9) In conventional drained axial compression/extension soil tests, Slope = Dp/Dq = 1/3.

(10) If $\alpha_c = 0.0$, the generation option 1 is used by default.

EXAMPLE

```
Stress_model  /
      material_name = multi_yield  /
      max_number_yield_surfaces = 20  /
      material_type = nonlinear

Material_set_number = 1  /
      shear_modulus = 3.00E7  /
      bulk_modulus = 2.00E7  /
      mass_density = 2.65E3  /
      fluid_mass_density = 1.E3  /
      fluid_bulk_modulus = 1.0E9  /
      porosity = 0.43  /
      plasticity_sub_type = 8  /
      ref_mean_stress = 2.0E5  /
      power_exponent = 0.5  /
      number_yield_surfaces = 20  /
      dilation_angle_comp = 30.0  /
      dilation_angle_ext = 30.0  /
      dilatational_parameter_Xpp = 1.0  /
      friction_angle_comp = 30.0  /
      friction_angle_ext = 30.0  /
      lateral_stress_coefficient = 1.0  /
      axial_stress_path_slope = 0.33  /
      max_shear_strain_comp = 0.05  /
      max_shear_strain_ext = 0.03  /
      initial_stress_11 = -2.E5  /
      initial_stress_22 = -2.E5  /
      initial_stress_33 = -2.E5
```

Notes . .

Notes . .

10.10 Stress_Model: Multi-Mechanism Elasto-PlasticModels (Geomaterials)

ISHIHARA

Material_name = ISHIHARA
m, nys(m), g1(m), b1(m), rhos(m), [alf(m), rhof(m), xnf(m)]
at(m), p1(m), xn(m), am(m), ar(m), ac(m)
fad(m), ptad(m), xmu0(m), sc(m)
s11(m), s22(m), s33(m), s12(m), s23(m), s31(m)
< for m = 1 , numat >

This constitutive model may be used with plane strain, axisymmetric and three dimensional options. *Finite deformation effects are not accounted for by this model*.

10.10.1 Material Properties Cards – (Numat sets))

Card 1 Elastic Parameters

Note	Variable	Default	Description
	M	[0]	Material set number ≤ Numat
	G(M)	[0.0]	Elastic shear modulus (solid phase)
	B(M)	[0.0]	Elastic bulk modulus (solid phase)
	RHOS(M)	[0.0]	Mass density (solid phase) per unit volume
(1)	ALF(M)	[0.0]	Bulk modulus (fluid phase)
(1)	RHOF(M)	[0.0]	Mass density (fluid phase) per unit volume
(1)	XNF(M)	[0.0]	Porosity

Notes/
(1) Only applicable to porous media models.

Card 2 Yield Parameters

Note	Variable	Default	Description
	At(M)	[0.0]	Attraction ≥ 0.0
	P1(M)	[0.0]	Reference mean normal stress ≥ 0.0
	XN(M)	[0.0]	Power exponent
	Am(M)	[1.4E-3]	Isotropic hardening parameter ≥ 0.0
	Ar(M)	[3.5E-5]	Kinematic hardening parameter ≥ 0.0
	Ac(M)	[6.1E-4]	Hardening parameter (liquefaction) ≥ 0.0

Card 3 Yield Parameters

Note	Variable	Default	Description
	FAD(M)	[0.0]	Friction angle in degrees
	PTAD(M)	[0.0]	Phase transformation angle in degrees
	XMU0(M)	[0.0]	Minimum value of dilatancy parameter
	Sc(M)	[3.5E-3]	Index constant ≥ 0.0

Card 4 Initial Stresses

Note	Variable	Default	Description
	S(1,M)	[0.0]	Component 11 (SIGMA11)
	S(2,M)	[0.0]	Component 22 (SIGMA22)
	S(3,M)	[0.0]	Component 33 (SIGMA33)
	S(4,M)	[0.0]	Component 12 (SIGMA12)
	S(5,M)	[0.0]	Component 23 (SIGMA23)
	S(6,M)	[0.0]	Component 31 (SIGMA31)

Notes/
(1) Tensile Stresses are positive.

Notes . .

Notes . .

10.11 Stress_Model: Phillips Constitutive Model

PHILLIPS

Material_name = PHILLIPS Number_of_uniaxial_curves = ncurve, etc...

Finite deformation effects are not accounted for by this model.

10.11.1 Material Control Data

Note	Variable Name	Type	Default	Description
	Number_of_uniaxial_curves	integer	[1]	Number of uniaxial curves ≥ 1
	Max_number_of_data_points	integer	[0]	Maximum number of data points ≥ 3
	Number_of_creep_curves	integer	[0]	Number of creep curves ≥ 0
	Max_number_of_creep_data	integer	[0]	Maximum number of creep data ≥ 1

10.11.2 Material Properties Data

Note	Variable Name	Type	Default	Description
	Material_set_number	integer	[1]	Material set number ≤ Numat
	Mass_density	real	[0.0]	Mass density ρ
	Shear_modulus	real	[0.0]	Shear modulus G
	Bulk_modulus	real	[0.0]	Bulk modulus B
	Friction_angle	real	[0.0]	Friction angle φ > 0.0
	Cohesion	real	[0.0]	Cohesive coefficient c ≥ 0.0
	Dilation_angle	real	[0.0]	Dilation angle ψ ≥ 0.0
(1)	Initial_stress			
	initial_stress_11	real	[0.0]	Component 11 (σ_{11})
	initial_stress_22	real	[0.0]	Component 22 (σ_{22})
	initial_stress_33	real	[0.0]	Component 33 (σ_{33})
	initial_stress_12	real	[0.0]	Component 12 (σ_{12})
	initial_stress_23	real	[0.0]	Component 23 (σ_{23})
	initial_stress_31	real	[0.0]	Component 31 (σ_{31})

(cont'd)

(cont'd)

Note	Variable Name	Type	Default	Description
	Solid_mass_density	real	[0.0]	Mass density (solid phase) ρ_s
	Fluid_mass_density	real	[0.0]	Mass density (fluid phase) ρ_w
	Fluid_bulk_modulus	real	[0.0]	Fluid bulk modulus λ^w
	Porosity	real	[0.0]	Porosity n^w
	K0_at_rest	real	[ν/(1-ν)]	Lateral stress coefficient at rest.
	K0_loading	real	[ν/(1-ν)]	Lateral stress coefficient upon loading.
	K0_unloading	real	[ν/(1-ν)]	Lateral stress coefficient upon unloading.
	Direction_of_anisotropy	integer	[3]	Direction of principal anisotropy. ≥ 1 and ≤ NSD
	Porosity_update on / **off**	list	[off]	Porosity update option
	Failure_check **on** / off	list	[on]	Failure option
	Critical_friction_angle	real	[0.0]	Critical friction angle $\varphi > 0.0$
	Primary_curve	integer	[1]	Primary uniaxial stress-strain curve N_1 ≥ 1 and ≤ Nuniaxial_curve
	Secondary_curve	integer	[N_1]	Secondary uniaxial stress-strain curve N_2 ≥ 1 and ≤ Nuniaxial_curve
	Interpolation_option on / **off**	list	[off]	Interpolation option
	Ref_fluid_pressure	real	[0.0]	Pore fluid pressure multiplier p^w
	Pressure_load_time	integer	[1]	Pore fluid pressure load time function number

(cont'd)

(cont'd)

Note	Variable Name	Type	Default	Description
	Saturation_multiplier	real	[0.0]	Pore fluid saturation multiplier
	Saturation_load_time	integer	[1]	Pore fluid saturation load time function number
	Critical_saturation	real	[0.0]	Pore fluid critical saturation S_c
	Ultimate_saturation	real	[0.0]	Pore fluid ultimate saturation S_u
	Ultimate_curve	integer	[N_2]	Ultimate uniaxial stress-strain curve $S_c < S < S_u$ ≥ 1 and $\leq$ Nuniaxial_curve
	Power_exponent	real	[0.0]	Power exponent (Creep)
	Creep_update on / **off**	list	[off]	Creep update
	Creep_load_time	integer	[1]	Creep load time function number
	Creep_curve	integer	[1]	Creep curve ≥ 1 and $\leq$ Ncreep_curve
(2)	File_name	string	[none]	File name (optional). Name must be enclosed in quotation marks.

Notes/
(1) Tensile stresses are positive.

(2) This option allows the uniaxial stress-strain data and creep data to be read in from another file.

10.11.3 Uniaxial Stress-Strain Curves and Creep Data

The uniaxial stress-strain data are first input followed by the creep data. Following are the required data records for input of the uniaxial stress-strain curves:

Record 1: Identifying name for the following uniaxial stress-strain data.

Record 2

Note	Variable	Default	Description
	N	[0]	Material curve set number ≥ 1 and ≤ Nuniaxial_curve
	NMAX(N)	[0]	Number of data points ≥ 3 and ≤ Nuniaxial_max
	SCX(N)	[0.0]	Scaling factor for strains ≥ 0.0
	SCY(N)	[0.0]	Scaling factor for stresses ≥ 0.0

Record(s) 3 (NMAX(N) sets)

Note	Variable	Default	Description
	STRAIN	[0.0]	Strain
	STRESS	[0.0]	Stress

The creep data follow the uniaxial data. Following are the required data records for input of the creep curves:

Record 1: Identifying name for the following creep data.

Record 2

Note	Variable	Default	Description
	N	[0]	Material curve set number ≥ 1 and ≤ Ncreep_curve
	NMAX(N)	[0]	Number of data points ≥ 3 and ≤ Ncreep_max
	SCX(N)	[0.0]	Scaling factor for stress ≥ 0.0
	SCY(N)	[0.0]	Scaling factor for R_0 ≥ 0.0

Record(s) 3 (NMAX(N) sets)

Note	Variable	Default	Description
	STRESS	[0.0]	Stress
	R0	[0.0]	Reference time resistance R_0
	r	[0.0]	Time resistance number r
	epsilon0	[0.0]	24 hrs strain at each load step $epsilon_0$ (in %)

Notes . .

10.12 Stress_Model: Isotropic Hypo Plastic Model

HYPO_PLASTIC

Material_name = HYPO_PLASTIC
Material_set_number = mset , etc...

The incremental constitutive equations for the hypoplastic model are written as:

$$d\boldsymbol{\sigma} = d\mathbf{s} + dp\boldsymbol{\delta}$$

where

$\boldsymbol{\sigma}$ = effective stress tensor
$\mathbf{s} = \boldsymbol{\sigma} - p\boldsymbol{\delta}$ = deviatoric stress
$p = \frac{1}{3} tr\boldsymbol{\sigma}$ = mean effective stress

The hypoplastic incremental equations are written as:

$$d\,\mathbf{s} = 2\,G\,d\,\boldsymbol{\gamma} \qquad\qquad dp = B\,d\,\varepsilon_v$$

where

$\boldsymbol{\varepsilon}$ = strain
$\boldsymbol{\gamma} = \boldsymbol{\varepsilon} - \frac{1}{3}\varepsilon_v\,\boldsymbol{\delta}$ = shear strain
$\varepsilon_v = t\,r\,\boldsymbol{\varepsilon}$ = volumetric deformation

The bulk and shear moduli depend on the current mean effective stress p according to:

$$G = G_1\left(\frac{p}{p_1}\right)^n \qquad\qquad B = B_1\left(\frac{p}{p_1}\right)^n$$

where p_1 is a reference mean stress, and n the power exponent.

The constitutive equations must be integrated between t_k and t_{k+1}. Integration for the mean stress is straightforward and gives:

$$\int_{p_k}^{p_{k+1}} \frac{dp}{p^n} = \frac{B_1}{p_1^n}\left[\varepsilon_v\right]_{t_k}^{t_{k+1}}$$

$$p_{k+1} = p_k \left[1 + (1-n) \frac{B_1}{p_k} \left(\frac{p_k}{p_1} \right)^n \Delta \varepsilon_v \right]^{\frac{1}{1-n}} \qquad \text{if } n \neq 1$$

$$p_{k+1} = p_k \exp\left(\frac{B_1}{p_1} \Delta \varepsilon_v \right) \qquad \text{if } n = 1$$

To integrate the deviatoric equation it is assumed that the shear strain tensor varies linearly between t_k and t_{k+1}. Then:

$$d\mathbf{s} = 2\, G(p) d\boldsymbol{\gamma} = 2\, G(p) \frac{d\boldsymbol{\gamma}}{dp} dp = 2\, G(p)\, \mathbf{a}_k dp$$

where $\mathbf{a}_k$ is the rate of increase of $d\boldsymbol{\gamma}$, assumed constant during the time interval from t_k to t_{k+1}.

$$\mathbf{a}_k = \frac{\boldsymbol{\gamma}_{k+1} - \boldsymbol{\gamma}_k}{p_{k+1} - p_k} = \frac{\Delta \boldsymbol{\gamma}}{p_{k+1} - p_k}$$

Integration of this equation gives:

$$\Delta \mathbf{s} = 2 \frac{1}{n+1} \frac{G_1}{p_1^n} \frac{p_{k+1}^{n+1} - p_k^{n+1}}{p_{k+1} - p_k} \Delta \boldsymbol{\gamma}$$

The equations are valid either for initial loading or for unloading-reloading. The only difference is that for unloading-reloading the values of G_1 and B_1 are replaced by G_1^u and B_1^u, respectively.

Initial loading and unloading-reloading states are identified by the following conditions:

Initial loading: $\dot{\varepsilon}_v \geq 0$, $p_k \geq p_{\max}$

Unloading: $\dot{\varepsilon}_v < 0$

Reloading: $\dot{\varepsilon}_v \geq 0$, $p_k < p_{\max}$

Where $p_{\max}$ is the maximum mean effective stress reached during the loading history, up to time t_k.

Note	Variable Name	Type	Default	Description
• *Keywords Read Method*				
	Material_set_number	integer	[1]	Material set number ≤ Numat
	Mass_density	real	[0.0]	Mass density ρ
	Shear_modulus	real	[0.0]	Shear modulus G_1
	Bulk_modulus	real	[0.0]	Bulk modulus B_1
	Unloading_shear_modulus	real	[0.0]	Unloading shear modulus G_1^u
	Unloading_bulk_modulus	real	[0.0]	Unloading bulk modulus B_1^u
	Ref_mean_stress	real	[0.0]	Ref mean stress p_1
	Power_exponent	real	[0.0]	Power exponent n
(1)	Initial_stress:			
	initial_stress_11	real	[0.0]	Component 11 (σ_{11})
	initial_stress_22	real	[0.0]	Component 22 (σ_{22})
	initial_stress_33	real	[0.0]	Component 33 (σ_{33})
	initial_stress_12	real	[0.0]	Component 12 (σ_{12})
	initial_stress_23	real	[0.0]	Component 23 (σ_{23})
	initial_stress_31	real	[0.0]	Component 31 (σ_{31})
(2)	Solid_mass_density	real	[0.0]	Mass density (Solid Phase) ρ_s
(2)	Fluid_mass_density	real	[0.0]	Mass density (Fluid Phase) ρ_w
(2)	Fluid_bulk_modulus	real	[0.0]	Fluid bulk modulus λ^w
(2)	Porosity	real	[0.0]	Porosity n^w

• *List Read Method*

Material data must follow in the form:

< m, G_1 (m), B_1 (m), G_1^u (m), B_1^u (m), p_1 (m), n (m), ρ(m), λ^w(m), ρ_w(m), n^w(m), Pf(m) >
< (Stres(i, m), i = 1, 6) >
< terminate with a blank record >.

Notes/

(1) Tensile stresses are positive.

(2) Only applicable to porous media models.

Notes . .

10.13 Stress_Model: Mohr_Coulomb Elasto-(Visco-)Plastic Model

MOHR_COULOMB

Material_name = MOHR_COULOMB
Material_set_number = mset , etc...

The yield function is of the following type:

$$f = \sqrt{\frac{3}{2} tr\underset{\sim}{s}^2} + \frac{6\sin\varphi}{3-\sin\varphi}(p-a)\,g(\theta)$$

where a = attraction = *c/tanφ, c=cohesion* and *φ=friction angle,*

$$g(\theta) = \frac{2M_k}{(1+M_k)-(1-M_k)\sin 3\theta}$$

in which

$$\sin 3\theta = -\sqrt{6}\bar{J}_3 / \bar{J}_2^{3/2}$$

$$\bar{J}_2 = tr\underset{\sim}{s}^2 \qquad \bar{J}_3 = tr\underset{\sim}{s}^3 \qquad \underset{\sim}{s} = \underset{\sim}{\sigma} - p\underset{\sim}{\delta} \qquad p = \frac{1}{3} tr\underset{\sim}{\sigma}$$

and $M_k =$ material parameter:

$$M_k = \frac{3-\sin\varphi}{3+\sin\varphi}$$

The following hyperelastic stored energy function with uncoupled volumetric and deviatoric parts is employed:

$$\psi = 1/2\,\mathrm{G}\left(I_b^e - 3\right) + U\left(J^e\right) \quad \text{with} \quad U\left(J^e\right) = B\left(J^e Ln J^e - J^e + 1\right)$$

where

$$J^e = \det\left(\boldsymbol{F}^e\right), \quad I_b^e = J^{-2/3}\boldsymbol{b}^e : \boldsymbol{g}, \quad \boldsymbol{b}^e = \boldsymbol{F}^e \boldsymbol{F}^{et}$$

Note	Variable Name	Type	Default	Description
• *Keywords Read Method*				
	Material_set_number	integer	[1]	Material set number ≤ Numat
(1)	Hyperelastic_case	integer	[0]	Hyperelastic free energy function: = 0, $U(J) = B(JLnJ - J + 1)$ = 1, $U(J) = B(LnJ)^2 \ / \ 2$
	Mass_density	real	[0.0]	Mass density ρ
	Shear_modulus	real	[0.0]	Shear modulus G
	Bulk_modulus	real	[0.0]	Bulk modulus B
	Activation_time	real	[0.0]	Time at which nonlinearities are activated.
	Friction_angle	real	[0.0]	Friction angle $\varphi > 0.0$
	Cohesion	real	[0.0]	Cohesive coefficient *c*
	Dilation_angle	real	[0.0]	Dilation angle $\psi \geq 0.0$
	Tension_cutoff on / **off**	list	[off]	Tension cutoff options
	Relaxation_time	real	[0.0]	Relaxation time constant $\eta \geq 0.0$ ≤ 0.0, Elastoplastic > 0.0, Elastoviscoplastic
	Variable_cohesion	integer	[0]	Variable cohesion load time function number
	Variable_friction	integer	[0]	Variable friction angle load time function number
(2)	Initial_stress			
	initial_stress_11	real	[0.0]	Component 11 (σ_{11})
	initial_stress_22	real	[0.0]	Component 22 (σ_{22})
	initial_stress_33	real	[0.0]	Component 33 (σ_{33})
	initial_stress_12	real	[0.0]	Component 12 (σ_{12})
	initial_stress_23	real	[0.0]	Component 23 (σ_{23})
	initial_stress_31	real	[0.0]	Component 31 (σ_{31})

(cont'd)

(cont'd)

Note	Variable Name	Type	Default	Description
(3)	Solid_mass_density	real	[0.0]	Mass density (solid phase) ρ_s
(3)	Fluid_mass_density	real	[0.0]	Mass density (fluid phase) ρ_w
(3)	Fluid_bulk_modulus	real	[0.0]	Fluid bulk modulus λ^w
(3)	Porosity	real	[0.0]	Porosity n^w

• *List Read Method*

Material data must follow in the form:

< m, IHyper(m), G(m), B(m), ρ(m), λ^w(m), ρ_w(m), n^w(m), Pf(m) >
< ϕ(m), c(m), ψ(m), η(m), ltime_coh(m), ltime_phi(m) >
< (Stres(i, m), i = 1, 6) >
< terminate with a blank record >.

Notes/

(1) Only applicable to finite deformation case (see Section 9.2.1).

(2) Tensile stresses are positive.

(3) Only applicable to porous media models.

Notes . .

10.14 Stress_Model: 1D Multi-Yield Model

STRESS_1D

Material_name = STRESS_1D
Material_set_number = mset , etc...

The Stress_1D model can only be used with structural beam and truss elements. It allows the axial-stress vs axial-strain to be described as linear or nonlinear (see Fig. 10.12). If nonlinear, a piecewise linear behavior is assumed and the behavior may be prescribed as either elastic or hysteretic.

Note	Variable Name	Type	Default	Description
• *Keywords Read Method*				
	Material_set_number	integer	[1]	Material set number ≤ Numat
	Mass_density	real	[0.0]	Mass density ρ
	Youngs_modulus	real	[0.0]	Young's modulus E
	Poissons_ratio	real	[0.0]	Poisson's ratio ν
	Modulus_coeff_0	real	[0.0]	Initial modulus coefficient H_0
	Material_type **elastic** / hysteretic	list	[elastic]	Material type selection
	Activation_time	real	[0.0]	Time at which nonlinearities are activated.
	Tension_cutoff on / **off**	list	[off]	Tension cutoff option
	Cutoff_tension	real	[0.0]	Cutoff in tension
	Tension_failure on / **off**	list	[off]	Tension failure option
	Compression_cutoff on / **off**	list	[off]	Compression cutoff option
	Cutoff_compression	real	[0.0]	Cutoff in compression
	Compression_failure on / **off**	list	[off]	Compression failure option
	Stress_level_i	real	[0.0]	Stress level $i (i \le 5)$
	Modulus_coeff_i	real	[0.0]	Modulus coefficient $H_i (i \le 5)$

(cont'd)

(cont'd)

Note	Variable Name	Type	Default	Description

• *List Read Method*

Material data must follow in the form:

< $m, H_0(m), \rho(m)$, mat_type (m), icut(m), cutoff_tension(m), cutoff_compression(m),

(stress_level (i, m), modulus_coeff (i, m), i = 1, 5) >

< terminate with a blank record >.

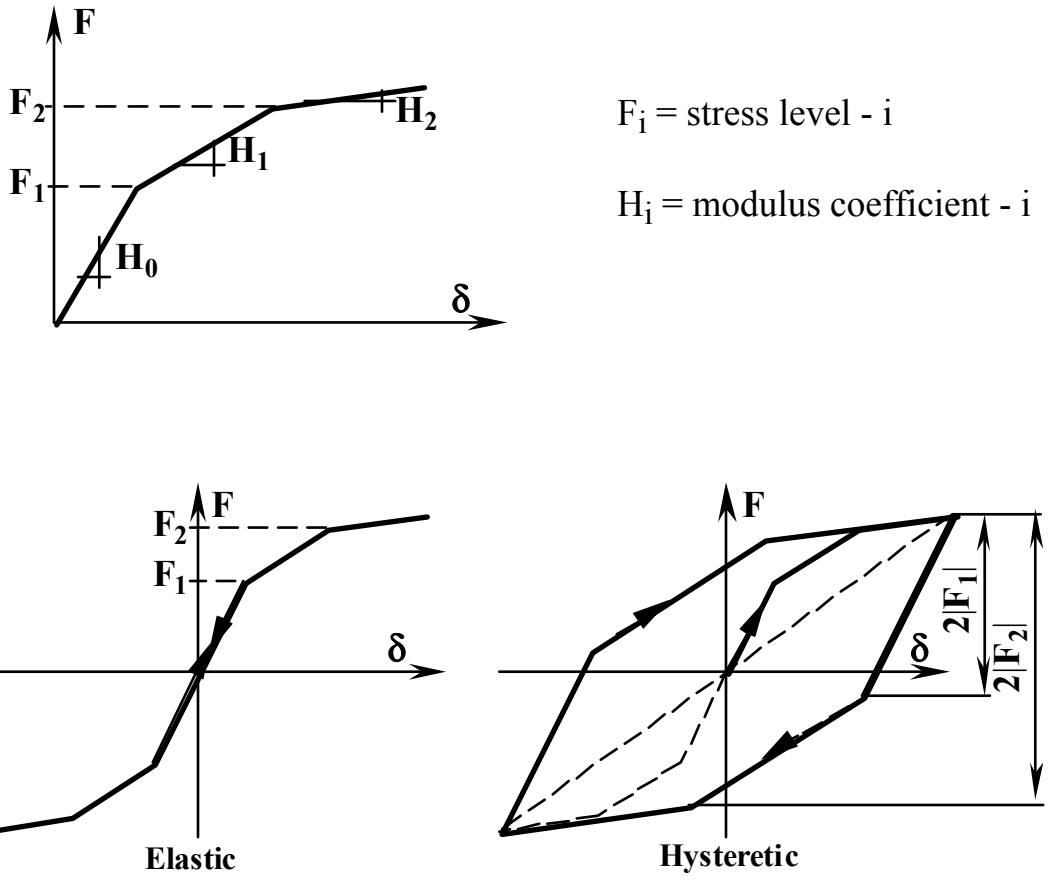


Figure 10.14. Nonlinear Stress Model

Notes . .

10.15 Stress_Model: Newtonian Fluid Model

NEWTONIAN_FLUID

Material_name = NEWTONIAN_FLUID
Material_set_number = mset , etc...

Note	Variable Name	Type	Default	Description
• *Keywords Read Method*				
	Material_set_number	integer	[1]	Material set number ≤ Numat
(1)	Bulk_modulus	real	[0.0]	Bulk modulus / penalty parameter λ
	Bulk_viscosity	real	[0.0]	Bulk viscosity λ^w
	Shear_viscosity	real	[0.0]	Shear viscosity μ^w
	Mass_density	real	[0.0]	Mass density ρ_w

• *List Read Method*

Material data must follow in the form:

< m, λ(m), λ^w(m), μ^w(m), ρ_w(m), Pf(m) >

< terminate with a blank record >.

Notes/

(1) For incompressible applications, $\lambda = c \max\left(\mu^w, \mu^w R_e\right)$ where $c = 10^7$ in machines with 60 to 64 bits floating point word lengths (see Section 9.2.0.5).

Notes . .

10.16 Heat_Conduction_Model: Generalized Heat Conduction Model

HEAT_CONDUCTION

Material_name = HEAT_CONDUCTION
Material_set_number = mset , etc...

The generalized heat conduction model is defined as follows.

Note	Variable Name	Type	Default	Description
	Material_set_number	integer	[1]	Material set number ≤ Numat
	Mass_density	real	[0.0]	Mass density ρ (matrix/bulk material)
	Reference_temperature	real	[273.15]	Reference temperature T_0
	Relaxation_time_1	real	[0.0]	First relaxation time t_1
	Relaxation_time_2	real	[0.0]	Second relaxation time t_2
	Specific_heat	real	[0.0]	Specific heat c (matrix/bulk material)
(1)	Coefficient_c1	real	[0.0]	Nonlinear Specific Heat Multiplier c_1
	Coefficient_c2	real	[0.0]	Nonlinear Specific Heat Multiplier c_2
	Coefficient_c3	real	[0.0]	Nonlinear Specific Heat Multiplier c_3
	Coefficient_c4	real	[0.0]	Nonlinear Specific Heat Multiplier c_4
	Coefficient_c5	real	[0.0]	Nonlinear Specific Heat Multiplier c_5
• *Thermal Conductivity*				
	Thermal_Conductivity	list	[none]	Thermal conductivity (matrix/bulk material)
	Type **isotropic** / anisotropic	list	[*]	Form of thermal conductivity matrix: if isotropic only k_{11} need be specified.
	k_11	real	[0.0]	Conductivity k_{11}
	k_22	real	[0.0]	Conductivity k_{22}
	k_33	real	[0.0]	Conductivity k_{33}
	k_12	real	[0.0]	Conductivity k_{12}
	k_23	real	[0.0]	Conductivity k_{23}
	k_13	real	[0.0]	Conductivity k_{13}
(2)	Coefficient_cd1	real	[0.0]	Nonlinear Conductivity Multiplier cd_1
	Coefficient_cd2	real	[0.0]	Nonlinear Conductivity Multiplier cd_2
	Coefficient_cd3	real	[0.0]	Nonlinear Conductivity Multiplier cd_3
	Coefficient_cd4	real	[0.0]	Nonlinear Conductivity Multiplier cd_4
	Coefficient_cd5	real	[0.0]	Nonlinear Conductivity Multiplier cd_5

(cont'd)

(cont'd)

Note	Variable Name	Type	Default	Description
• *Thermal Coefficient*				
	Thermal_coefficient	list	[none]	Thermal Moduli
	Type **isotropic** / anisotropic	list	[*]	Form of thermal coefficient matrix: if isotropic only β_{11} need be specified.
(3)	Beta_11	real	[0.0]	Thermal Coefficient β_{11}
	Beta_22	real	[0.0]	Thermal Coefficient β_{22}
	Beta_33	real	[0.0]	Thermal Coefficient β_{33}
	Beta_12	real	[0.0]	Thermal Coefficient β_{12}
	Beta_23	real	[0.0]	Thermal Coefficient β_{23}
	Beta_13	real	[0.0]	Thermal Coefficient β_{13}
(4)	Coefficient_cb1	real	[0.0]	Nonlinear Thermal Multiplier cb_1
	Coefficient_cb2	real	[0.0]	Nonlinear Thermal Multiplier cb_2
• *Thermal Expansion Coefficient*				
	Thermal_Expansion	list	[none]	Thermal Expansion Moduli
	Type **isotropic** / anisotropic	list	[*]	Form of thermal coefficient matrix: if isotropic only α_{11} need be specified.
(5)	Alpha_11	real	[0.0]	Thermal Expansion α_{11}
	Alpha_22	real	[0.0]	Thermal Expansion α_{22}
	Alpha_33	real	[0.0]	Thermal Expansion α_{33}
	Alpha_12	real	[0.0]	Thermal Expansion α_{12}
	Alpha_23	real	[0.0]	Thermal Expansion α_{23}
	Alpha_13	real	[0.0]	Thermal Expansion α_{13}
(4)	Coefficient_cb1	real	[0.0]	Nonlinear Thermal Multiplier cb_1
	Coefficient_cb2	real	[0.0]	Nonlinear Thermal Multiplier cb_2

EXAMPLE

```
Heat_conduction_model  /
   material_type = Linear  /
   material_name = heat_conduction
      material_set_number = 1  /
      reference_temperature = 1.0  /
      mass_density = 1.0  /
      specific_heat = 1.0e-6  /
      thermal_conductivity  /
         type = isotropic  /
         k_11 = 1.60e-3  /
      thermal_coefficient  /
         type = isotropic  /
         beta_11 = 1.00
```

Notes/

(1) The nonlinear specific heat c(T) is computed by the following equation:

$$c(T) = c\left(1 + c_1 T + c_2 / T + c_3 / T^2 + c_4 / T^3 + c_5 / T^4\right)$$

where T = $(\theta + \mathrm{T}_0)$ = absolute temperature.

(2) The nonlinear thermal conductivity **k**(T) is computed by the following equation:

$$\mathbf{k}(T) = \mathbf{k}\left(1 + cd_1 T + cd_2 T^2 + cd_3 T^3 + cd_4 T^4 + cd_5 T^5\right)$$

where T = $(\theta + \mathrm{T}_0)$ = absolute temperature.

(3) In the linear isotropic case, the thermal modulus β is related to the usual coefficient of thermal expansion α via:

$$\beta = \frac{\mathrm{E}}{1 - 2\nu}\alpha$$

where E = Young's modulus, and ν = Poisson's ratio.

(4) The nonlinear thermal modulus $\beta(\mathrm{T})$ is computed by the following equation:

$$\beta(\mathrm{T}) = \beta\left(1 + \mathrm{cb}_1 \mathrm{T} + \mathrm{cb}_2 \mathrm{T}^2\right)$$

where T = $(\theta + \mathrm{T}_0)$ = absolute temperature.

(5) The thermal moduli β are computed from the thermal expansion moduli α via:

$$\beta = \mathbf{E} : \alpha$$

where **E** denotes the linear elasticity tensor.

10.16.1 Multi-Phase Fluid Flow

For multi-phase fluid flow problems (see Section 9.2.0.16), the following material data must also be provided for each phase.

Note	Variable Name	Type	Default	Description
	Phase_number	integer	[1]	Phase number
	Specific_heat	real	[0.0]	Specific heat C_P
•	*Thermal Conductivity*			
	Thermal conductivity	list	[none]	Thermal conductivity
	Type **isotropic** / anisotropic	list	[*]	Form of thermal conductivity matrix: if isotropic only k_{11} need be specified.
	k_11	real	[0.0]	Conductivity k_{11}
	k_22	real	[0.0]	Conductivity k_{22}
	k_33	real	[0.0]	Conductivity k_{33}
	k_12	real	[0.0]	Conductivity k_{12}
	k_23	real	[0.0]	Conductivity k_{23}
	k_13	real	[0.0]	Conductivity k_{13}

Notes . .

10.17 Scalar_Diffusion_Model: Generalized Scalar Diffusion Model

SCALAR_DIFFUSION

Material_name = SCALAR_DIFFUSION
Material_set_number = mset , etc...

The generalized scalar diffusion model is defined as follows.

Note	Variable Name	Type	Default	Description
	Material_set_number	integer	[1]	Material set number ≤ Numat
	Matrix_mass_density	real	[0.0]	Matrix mass density ρ_m
	Matrix_compressibility	real	[0.0]	Matrix compressibility C_m
	Grains_compressibility	real	[0.0]	Grains compressibility C_s
	Fluid_mass_density	real	[0.0]	Mass density ρ (fluid phase)
	Fluid_compressibility	real	[0.0]	Compressibility $\frac{1}{\rho}\frac{\partial \rho}{\partial p}$ [LT^2 / M]
	Fluid_viscosity	real	[0.0]	Viscosity μ [M / L T]
(1)	Ideal_fluid on / **off**	list	[off]	Ideal fluid/gas option
	Reference_pressure	real	[0.0]	Reference pressure
	Reference_temperature	real	[0.0]	Reference temperature
	Molecular_mass	real	[0.0]	Molecular mass of fluid/gas
	Porosity	real	[0.0]	Porosity
	Material_type **linear** / nonlinear	list	[linear]	Material type
	Number_of_phases	integer	[1]	Number of phases; Nphase
	Max_number_of_data_points	integer	[0]	Maximum number of data points used to define relative permeability/capillary pressure

(cont'd)

(cont'd)

Note	Variable Name	Type	Default	Description
• *Permeability*				
	Permeability	list	[none]	Permeability
	Type **isotropic** / anisotropic	list	[*]	Form of permeability matrix: if isotropic only k_{11} need be specified.
(2)	Name **conductivity** mobility intrinsic	list	[*]	Name of permeability: hydraulic conductivity [L / T] mobility [L^3 T / M] intrinsic permeability [L^2]
	k_11	real	[0.0]	Permeability k_{11}
	k_22	real	[0.0]	Permeability k_{22}
	k_33	real	[0.0]	Permeability k_{33}
	k_12	real	[0.0]	Permeability k_{12}
	k_23	real	[0.0]	Permeability k_{23}
	k_13	real	[0.0]	Permeability k_{13}
(3)	Exponent_porosity	real	[0.0]	Porosity exponent
• *Diffusivity / Dispersivity*				
	Diffusivity	list	[none]	Diffusivity
	Type **isotropic** / anisotropic	list	[*]	Form of diffusivity matrix: if isotropic only k_{11} need be specified.
	k_11	real	[0.0]	Diffusivity k_{11}
	k_22	real	[0.0]	Diffusivity k_{22}
	k_33	real	[0.0]	Diffusivity k_{33}
	k_12	real	[0.0]	Diffusivity k_{12}
	k_23	real	[0.0]	Diffusivity k_{23}
	k_13	real	[0.0]	Diffusivity k_{13}

EXAMPLE

```
Scalar_Diffusion_Model  /
   material_type = linear  /
   material_name = scalar_diffusion

      material_set_number = 1  /
      mass_density = 1.e3  /
      porosity = 0.30  /
      compressibility = 1.e3-6  /
      permeability  /
         type = isotropic  /
         name = conductivity  /
                k_11 = 1.60e-3
```

Notes /

(1) The equation of state is then of the form:

$$p = \rho \frac{R}{w} T$$

where p = pressure; ρ = mass density; R = fluid/gas constant (=8314 J/(kmol°K)); w = molecular mass; and T = temperature [°K]. Then, for instance for air:

w = 28.97 kg/kmol, and:

$$p\,[Pa] = \rho\,[kg/m^3] \times 286.987 \times T\,[^\circ\mathrm{K}]$$

(2) Let k denote the intrinsic permeability (units: [L^2]). Then

$$\frac{k}{\mu}\rho g = \text{hydraulic conductivity [L / T]}$$

$$\frac{k}{\mu} = \text{mobility [}L^3\text{ T / M]}$$

where μ = viscosity [M / L T], ρ = fluid mass density [M / L^3]; and g = acceleration of gravity [L / T^2].

(3) The permeability is function of porosity as:

$$k = k_0 \left(\frac{n}{n_0} \right)^{\alpha}$$

where n_0 = initial porosity, and α = porosity exponent.

10.17.1 Multi-Phase Fluid Flow

For multi-phase fluid flow problems the following material data must also be provided.

Note	Variable Name	Type	Default	Description
	eos_options PU_cmi tough2 Peng_Robinson dry_gas	list	[none]	Equation of state options
	reference_pressure	real	[0.0]	reference pressure
	reference_temperature	real	[0.0]	reference temperature
	matrix_mass_density	real	[0.0]	Matrix mass density $(\rho_m \geq 0)$
	matrix_compressibility	real	[0.0]	Matrix compressibility $(C_m \geq 0)$
	grains_compressibility	real	[0.0]	Grains compressibility $(C_s \geq 0)$
(1)	Relative_permeability **Touma_Vauclin** Linear Power Corey Grant perfect_mobility Fatt_Klikoff vanGenuchten_Mualem Verma Berea Modified_Corey Stone_3_phase	list	[*]	Relative permeability formula $k_{r\beta}$
(1)	rp_i	real	[0.0]	Coefficient RP(i)

(cont'd)

(cont'd)

Note	Variable Name	Type	Default	Description
(2)	Capillary_pressure Touma_Vauclin Linear Pickens Trust Milly Leverett vanGenuchten **none**	list	[none]	Capillary pressure formula
(2)	cp_i	real	[0.0]	Coefficient CP(i)

Notes /

(1) Relative Permeability Functions

IRP = 0 Touma_Vauclin function

The relative permeability is assumed in this case to be given by a curve fit to the experimental data as:

formula_type = 1 $k_{ri} = A_i (S_i)^{B_i}$

formula_type = 2 $k_{ri} = \dfrac{A_i}{A_i + \left(\dfrac{p_c}{p_o} \right)^{B_i}}$

where $p_c = p_2 - p_1$ = capillary pressure; p_0 = normalizing pressure; and S_i = degree of saturation.

IRP = 1 linear function

k_{rl} increases linearly from 0 to 1 in the range RP(1) $\le S_l \le$ RP(3);
k_{rg} increases linearly from 0 to 1 in the range RP(2) $\le S_g \le$ RP(4).
Restrictions: RP(3) > RP(1); RP(4) > RP(2).

IRP = 2 Power function

$k_{rl} = S_l$**RP(1)
$k_{rg} = 1$.

IRP = 3 Corey's curves (1954)

$k_{rl} = \hat{S}^4$
$k_{rg} = (1 - \hat{S})^2 (1 - \hat{S}^2)$

where $\hat{S} = (S_l - S_{lr}) / (1 - S_{lr} - S_{gr})$
with S_{lr} = RP(1); S_{gr} = RP(2)
Restrictions: RP(1) + RP(2) < 1.

IRP = 4 Grant's curves (Grant, 1977)

$k_{rl} = \hat{S}^4$
$k_{rg} = 1 - k_{rl}$
where $\hat{S} = (S_l - S_{lr}) / (1 - S_{lr} - S_{gr})$
with S_{lr} = RP(1); S_{gr} = RP(2)
Restrictions: RP(1) + RP(2) < 1.

IRP = 5 all phases perfectly mobile

$k_{rg} = k_{rl} = 1$ for all saturations; no parameters

IRP = 6 functions of Fatt and Klikoff (1959)

$k_{rl} = (S^*)^3$

$k_{rg} = (1 - S^*)^3$

where $S^* = (S_l - S_{lr}) / (1 - S_{lr})$

with S_{lr} = RP(1).

Restriction: RP(1) < 1.

IRP = 7 van Genuchten-Mualem model (Mualem, 1976; van Genuchten, 1980)

$$k_{rl} = \begin{cases} \sqrt{S^*}\left\{1 - \left(1 - [S^*]^{1/\lambda}\right)^{\lambda}\right\}^2 & \text{if } S_l < S_{ls} \\ 1 & \text{if } S_l \geq S_{ls} \end{cases}$$

Gas relative permeability can be chosen as one of the following two forms, the second of which is due to Corey (1954)

$$k_{rg} = \begin{cases} 1 - k_{rl} & \text{if } S_{gr} = 0 \\ \left(1-\hat{S}\right)^2 \left(1 - \hat{S}^2\right) & \text{if } S_{gr} > 0 \end{cases}$$

subject to the restriction $0 \leq k_{rl}, k_{rg} \leq 1$

Here, $S^* = (S_l - S_{lr}) / (S_{ls} - S_{lr})$, $\hat{S} = (S_l - S_{lr}) / (1 - S_{lr} - S_{gr})$

Parameters: RP(1) = λ
RP(2) = S_{lr}
RP(3) = S_{ls}
RP(4) = S_{gr}

Notation: Parameter λ is m in van Genuchten's notation, with m = 1 – 1/n; parameter n is often written as β.

IRP = 8 function of Verma et al. (1985)

$k_{rl} = \hat{S}^3$

$k_{rg} = A + B\,\hat{S} + C\,\hat{S}^2$

where $\hat{S} = (S_l - S_{lr}) / (S_{ls} - S_{lr})$,

Parameters as measured by Verma et al. (1985) for steam-water flow in an unconsolidated sand:

S_{lr} = RP(1) = 0.2 B = RP(4) = -1.7615

S_{ls} = RP(2) = 0.895 C = RP(5) = 0.5089

A = RP(3) = 1.259

IRP = 12 modified Corey function

$$k_{rl} = k_{rl}^{\max} \frac{\left(S_l - S_{lr}\right)^2}{\left(1 - S_{lr} - S_{gr}\right)^2}$$

$$k_{rg} = k_{rg}^{\max} \frac{\left(1 - S_l - S_{lr}\right)^2}{\left(1 - S_{lr} - S_{gr}\right)^2}$$

$$S_{lr} = RP(1) \qquad S_{gr} = RP(2)$$

$$k_{lr}^{\max} = RP(3) \qquad k_{rg}^{\max} = RP(4)$$

IRP = 14 Stone 3-phase model

a. Aqueous phase:

$$k_{ra} = \left(\frac{S_a - S_{ar}}{1 - S_{ar}} \right)^m$$

with S_{ar} = irreducible aqueous phase saturation (typically $S_{ar} = 0.15$)
m = exponent (typically m=3).

b. Liquid phase:

$$k_{rl} = \left[\frac{\hat{S} - S_a}{\hat{S} - S_{ar}} \right] \left[\frac{1 - S_{ar} - S_{lr}}{1 - S_a - S_{lr}} \right] \times \left[\frac{\left(\hat{S} - S_{ar}\right)\left(1 - S_a\right)}{\left(1 - S_{ar}\right)} \right]^m$$

$$\hat{S} = 1 - S_g - S_{lr}$$

S_{lr} = irreducible liquid phase saturation (typically $S_{lr} = 0.05$).

c. Gas phase:

$$k_{rg} = \left(\frac{S_g - S_{gr}}{1 - S_{ar}} \right)^m$$

S_{gr} = irreducible gas phase saturation (typically $S_{gr} = 0.01$).

Parameters: RP(1) = m
RP(2) = S_{ar}
RP(3) = S_{lr}
RP(4) = S_{gr}

(2) Capillary Pressure Functions

ICP = 0 Touma_Vauclin function
The capillary pressure vs saturation is assumed in this case to be given by a van Genuchten-type curve fit to the experimental data as:

$$S_l = \frac{S_{ls} - S_{lr}}{\left| 1 + \left(\alpha \frac{P_{cap}}{p_0} \right)^n \right|^{1-1/n}} + S_{lr}$$

where $P_{cap} = p_2 - p_1$ = capillary pressure; p_0 = normalizing pressure; and S_l = degree of saturation.

Parameters: S_{lr} = CP(1) S_{ls} = CP(2) n = CP(3) α = CP(4) p_0 = CP(5)

ICP = 1 linear function

$$P_{cap} = \begin{cases} +\text{CP(1)} & \text{for } S_l \le \text{CP(2)} \\ 0 & \text{for } S_l \ge \text{CP(3)} \\ +\text{CP(1)} \dfrac{\text{CP(3)} - S_l}{\text{CP(3)} - \text{CP(2)}} & \text{for CP(2)} < S_l < \text{CP(3)} \end{cases}$$

Restriction: CP(3) > CP(2).

ICP = 2 function of Pickens et al. (1979)

$$P_{cap} = +P_0 \left\{ ln\left[\frac{A}{B}\right]\left(1 + \sqrt{1 - B^2/A^2}\right)\right\}^{1/x}$$

with
$A = (1 + S_l/S_{l0})\,(S_{l0} - S_{lr}) / (S_{l0} + S_{lr})$
$B = 1 - S_l/S_{l0}$

where
P_0 = CP(1) S_{lr} = CP(2) S_{l0} = CP(3) x = CP(4)

Restrictions: $0 <$ CP(2) $< 1 \le$ CP(3); CP(4) $\neq 0$

ICP = 3 TRUST capillary pressure (Narasimhan et al., 1978)

$$P_{cap} = \begin{cases} + P_e + P_0 \left[\dfrac{1 - S_l}{S_l - S_{lr}} \right]^{1/\eta} & \text{for } S_l < 1 \\ 0 & \text{for } S_l > 1 \end{cases}$$

where
P_0 = CP(1) S_{lr} = CP(2) η = CP(3) P_e = CP(4)

Restrictions: CP(2) ≥ 0; CP(3) $\neq 0$

ICP = 4 Milly's function (Milly, 1982)

P_{cap} = +97.783 x 10^{A}

with

$$A = 2.26 \left(\frac{0.371}{S_l - S_{lr}} - 1 \right)^{1/4}$$

where S_{lr} = CP(1)
Restriction: CP(1) ≥ 0.

ICP = 6 Leverett's function (Leverett, 1941; Udell and Fitch, 1985)

$$P_{cap} = +P_0 \bullet \sigma(T) \bullet f(S_l)$$

with
$\sigma(T)$ = surface tension of water (supplied internally)
$f(S_l)$ = 1.417 $(1 - S^*)$ – 2.120 $(1 - S^*)^2$ + 1.263 $(1 - S^*)^3$

where
$S^* = (S_l - S_{lr}) / (1 - S_{lr})$

Parameters: P_0 = CP(1) S_{lr} = CP(2)
Restriction: $0 \leq$ CP(2) < 1

ICP = 7 van Genuchten function (van Genuchten, 1980)

$$P_{cap} = +P_0 \left([S^*]^{-1/\lambda} - 1 \right)^{1-\lambda}$$

subject to the restriction
$0 \leq P_{cap} \leq P_{max}$

Here,
$S^* = (S_l - S_{lr}) / (S_{ls} - S_{lr})$

Parameters: CP(1) = λ = l – l/n
CP(2) = S_{lr} (should be chosen smaller than the corresponding parameter in the relative permeability function; see note below.)
CP(3) = P_0

CP(4) = P_{max}
CP(5) = S_{ls}

Notation: Parameter λ is m in van Genuchten's notation, with m = l – l/n; parameter n is often written as β.

Note on parameter choices: In van Genuchten's derivation (1980), the parameter S_{lr} for irreducible water saturation is the same in the relative permeability and capillary pressure functions. As a consequence, for $S_l \rightarrow S_{lr}$ we have $k_{rl} \rightarrow 0$ and $P_{cap} \rightarrow -\infty$, which is unphysical because it implies that the radii of capillary menisci go to zero as liquid phase is becoming immobile (discontinuous). Accordingly, we recommend to always choose a smaller S_{lr} for the capillary pressure as compared to the relative permeability function.

ICP = 8 no capillary pressure
P_{cap} = 0 for all saturations; no parameters

10.17.1.1 Fluid Phase Data

Note	Variable Name	Type	Default	Description
(1)	Phase_number	integer	[1]	Phase number; i ≤ Nphase
	Phase_type **liquid** gas	list	[liquid]	Phase type
	Phase_name	string	[none]	Phase name; name(s) must be enclosed in quotation marks.
	Mass_density	real	[0.0]	Mass density ρ_i
	Compressibility	real	[0.0]	Compressibility $\frac{1}{\rho_i}\frac{\partial \rho_i}{\partial p_i}$ [LT^2 / M]
	Viscosity	real	[0.0]	Viscosity μ_i □ □M / L T□
	Saturation	real	[0.0]	Saturation S_i
	Minimum_saturation	real	[0.0]	Minimum saturation S_{ir}
	Maximum_saturation	real	[0.0]	Maximum saturation S_{is}
	Mass_fraction_dissolved_solid	real	[0.0]	mass fraction of total dissolved solid (TDS)
(2)	formula_type	integer	[iphase]	Relative permeability formula type (Touma_Vauclin option)
(2)	a_coefficient	real	[0.0]	Coefficient A_i in curve fit formula
(2)	b_coefficient	real	[0.0]	Coefficient B_i in curve fit formula
(2)	Normalizing_pressure	real	[0.0]	Normalizing pressure p_o in curve fit formula
• *Diffusivity / Dispersivity*				
	Diffusivity	list	[none]	Diffusivity
	Type **isotropic** / anisotropic	list	[*]	Form of diffusivity matrix: if isotropic only k_{11} need be specified.
	k_11	real	[0.0]	Diffusivity k_{11}
	k_22	real	[0.0]	Diffusivity k_{22}
	k_33	real	[0.0]	Diffusivity k_{33}
	k_12	real	[0.0]	Diffusivity k_{12}
	k_23	real	[0.0]	Diffusivity k_{23}
	k_13	real	[0.0]	Diffusivity k_{13}

Notes/
(1) The wetting phase **must** be defined as phase_number = 1.

(2) The relative permeability is assumed in this case to be given by a curve fit to the experimental data as:

formula_type = 1 $$k_{ri} = A_i \left(S_i\right)^{B_i}$$

formula_type = 2 $$k_{ri} = \frac{A_i}{A_i + \left(\frac{p_c}{p_o}\right)^{B_i}}$$

where $p_c = p_2 - p_1$ = capillary pressure; p_0 = normalizing pressure; and S_i = degree of saturation.

References / Bibliography

1. Touma, J. and M. Vauclin, "Experimental and Numerical Analysis of Two-Phase Infiltration in a Partially Saturated Soil," *Transport in Porous Media*, Vol. 1, 1986, pp. 27-55.

10.17.1.2 Relative Permeability and Capillary Pressure Data

Note	Variable Name	Type	Default	Description
	Material_set_number	integer	[1]	Material set number ≤ Numat
	Data_type Relative_permeability Capillary_pressure	list	[none]	Data type: Relative permeability Capillary pressure

• *Relative Permeability Case*

Data must follow in the form:

(1) < S_1, $k_{r1}(S_1)$, $k_{r2}(S_1)$ >

< etc..., terminate with a blank record >

• *Capillary Pressure Case*

Data must follow in the form:

(2) < S_1, $p_{c1}(S_1)$, $p_{c2}(S_1)$ >

< etc..., terminate with a blank record >

Notes /

(1) S_1 = degree of saturation for phase 1; $S_2 = 1 - S_1$

$k_{r1}(S_1)$ = relative permeability for phase 1

$k_{r2}(S_1)$ = relative permeability for phase 2

(2) $p_{c1}(S_1)$ = capillary pressure for phase 1 invasion

$p_{c2}(S_1)$ = capillary pressure for phase 1 drainage

EXAMPLE

```
Scalar_diffusion_model  /
  material_type = linear  /
  material_name = scalar_diffusion  /
  number_of_phases = 2  /
  max_number_of_data_points = 11
  material_set_number = 1  /
    porosity = 0.30  /
    permeability  /
      type = isotropic  /
      name = intrinsic  /
      k_11 = 2.1248e-11

  phase_number = 1  /
    mass_density = 62.4  /
    compressibility = 1.0e-6  /
    viscosity = 2.088543e-5

  phase_number = 2  /
    mass_density = 49.92  /
    compressibility = 1.0e-6  /
    viscosity = 8.3541723e-5

  material_set_number = 1  /
    data_type = Relative_permeability

        0.20      0.00      0.60
        0.25      0.02      0.47
        0.30      0.04      0.38
        0.35      0.07      0.31
        0.40      0.09      0.25
        0.45      0.13      0.18
        0.50      0.17      0.13
        0.55      0.22      0.09
        0.60      0.28      0.05
        0.65      0.35      0.02
        0.70      0.45      0.00

  material_set_number = 1  /
    data_type = Capillary_pressure

        0.200    208.854    208.854
        0.225    173.349    173.349
        0.250    148.287    148.287
        0.300    121.136    121.136
        0.350    102.339    102.339
        0.400     87.719     87.719
        0.500     68.922     68.922
        0.650     52.213     52.213
        0.700     48.663     48.663
```

Notes . .

10.18 Electric_Model: Generalized Electric Model

ELECTRIC

Material_name = ELECTRIC
Material_set_number = mset , etc...

The generalized electric model is defined as follows.

Note	Variable Name	Type	Default	Description
	Material_set_number	integer	[1]	Material set number ≤ Numat
	Material_type **linear** / nonlinear	list	[linear]	Material type: Restricted to linear in current implementation.
• *Dielectric Constants*				
	Permittivity	list	[none]	Permittivity
	Type **isotropic** / anisotropic	list	[*]	Form of permittivity matrix: If isotropic only k_{11} need be specified.
	k_11	real	[0.0]	Permittivity k_{11}
	k_22	real	[0.0]	Permittivity k_{22}
	k_33	real	[0.0]	Permittivity k_{33}
	k_12	real	[0.0]	Permittivity k_{12}
	k_23	real	[0.0]	Permittivity k_{23}
	k_13	real	[0.0]	Permittivity k_{13}
• *Piezoelectric Constants*				
	Piezoelectric_constants	list	[none]	Piezoelectric constants
	e_11	real	[0.0]	Piezoelectric constant e_{11}
	e_21	real	[0.0]	Piezoelectric constant e_{21}
	e_31	real	[0.0]	Piezoelectric constant e_{31}
	e_12	real	[0.0]	Piezoelectric constant e_{12}
	.	.	.	.
	etc.	.	.	.
	.	.	.	.
	e_36	real	[0.0]	Piezoelectric constant e_{36}

EXAMPLE

```
Define_Material_Model  /
   name = "dummy"  /
   number_of_material_sets = 1

Electric_Model  /
   material_type = linear  /
   material_name = electric

      material_set_number = 1  /
      permittivity  /
         type = isotropic  /
         k_11 = 1.00e+3  /
      piezoelectric_constants  /
         e_31 = 31, e_32 = 31, e_33 = 33, e_24 = 15, e_15 = 15
```

Notes . .

11.0 ELEMENT CONNECTIVITY DATA

NODAL_CONNECTIVITY

NODAL_CONNECTIVITY	file_name = "<string>" n, mat (n), (ien(i,n), i = 1, nen), ng < etc..., terminate with a blank record >

Define the connectivity. Various generation schemes are available depending on NEN the number of nodes used to define the element. These are detailed hereafter.

11.1 Element Nodal Data

Note	Variable	Default	Description
(1)	N	[0]	Element number
(2)	MAT(N)	[1]	Geometric/material properties set number
	IEN(1,N)	[0]	Number of 1st node
	IEN(2,N)	[0]	Number of 2nd node
	.	.	.
	etc.	.	.
	.	.	.
	IEN(NEN,N)	[0]	Number of NENth node
(3)	NG	[0]	Generation increment = 0, no generation ≠ 0, generate data

Notes/
(1) All elements must be read in on an element data record or generated. *Terminate with a blank record.*

(2) For Structural elements (see Section 9.3) for which both geometric and material properties are assigned to each element N:

MAT(N) = Material_set_number + Numel * Geometric_set_number

where Numel = Number_of_elements.

(3) If the generation parameter is not equal to 0, a generation data record must be input next. Various generation scheme are available depending on NEN the number of nodes used to define the element. These are detailed hereafter.

11.2 Element Generation Data

11.2.1 Element Generation Data for 1D Line Elements (NEN = 1, 2, or 3)

Note	Variable	Default	Description
	NEL (1)	[0]	Number of elements ≥ 0
	INCEL (1)	[0]	Element number increment; if = 0, set internally to 1
	INC (1)	[0]	Node number increment; if = 0, set internally to 1

11.2.2 Element Generation Data for 2D Plane Elements (NEN = 3 or 4)

Element nodes must be listed in *counterclockwise* order (see Figure 9.2.0.1). See Figure 11.2.2.1 for a schematic representation of the generation scheme.

Note	Variable	Default	Description
	NEL (1)	[0]	Number of elements in direction 1 ≥ 0
	INCEL (1)	[0]	Element number increment for direction 1; if = 0, set internally to 1
	INC (1)	[0]	Node number increment for direction 1; if = 0, set internally to 1
	NEL (2)	[0]	Number of elements in direction 2 ≥ 0
	INCEL (2)	[0]	Element number increment for direction 2; if = 0, set internally to NEL(1)
	INC (2)	[0]	Node number increment for direction 2; if = 0, set internally to (1 + NEL(1))*INC(1)

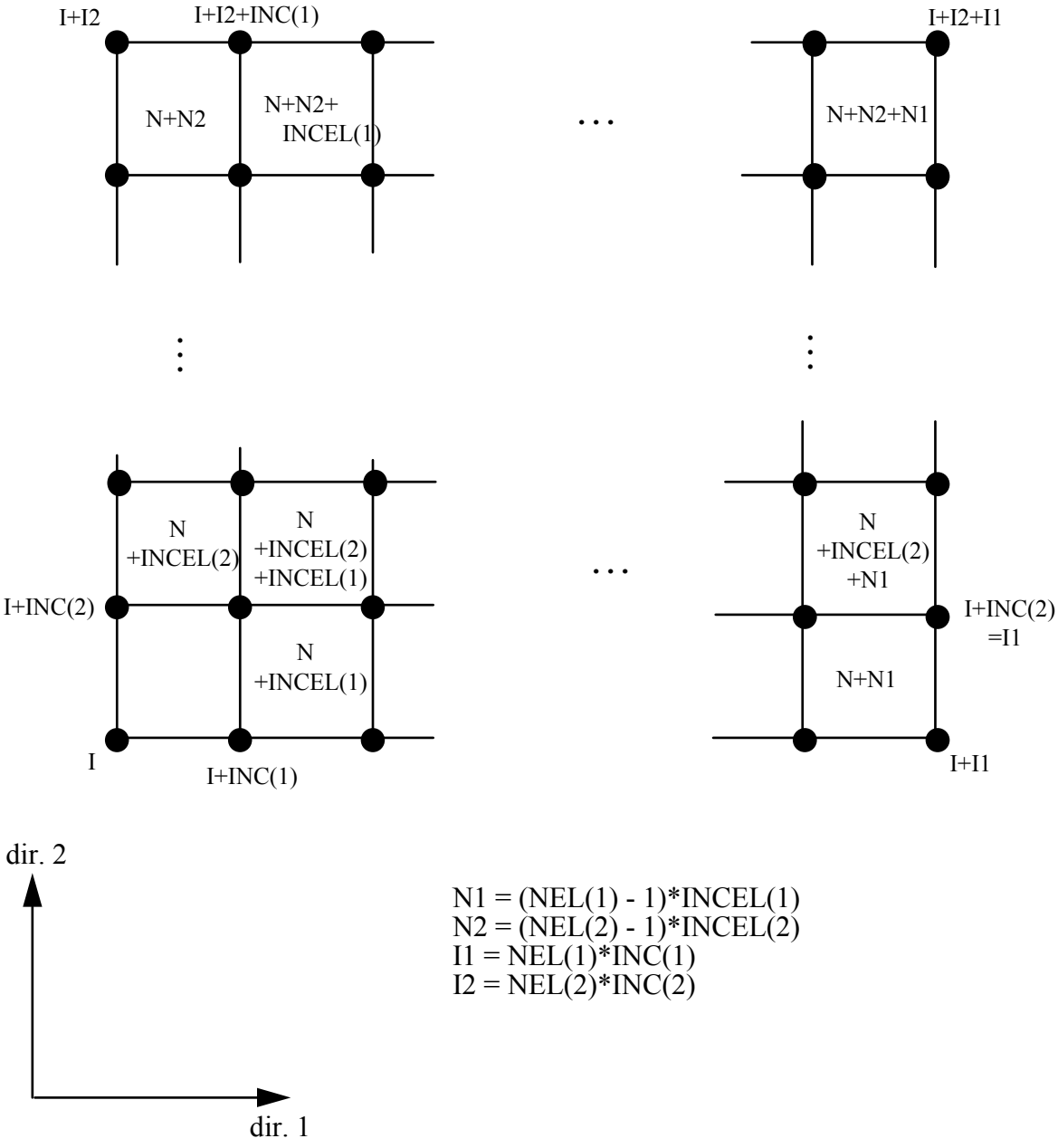
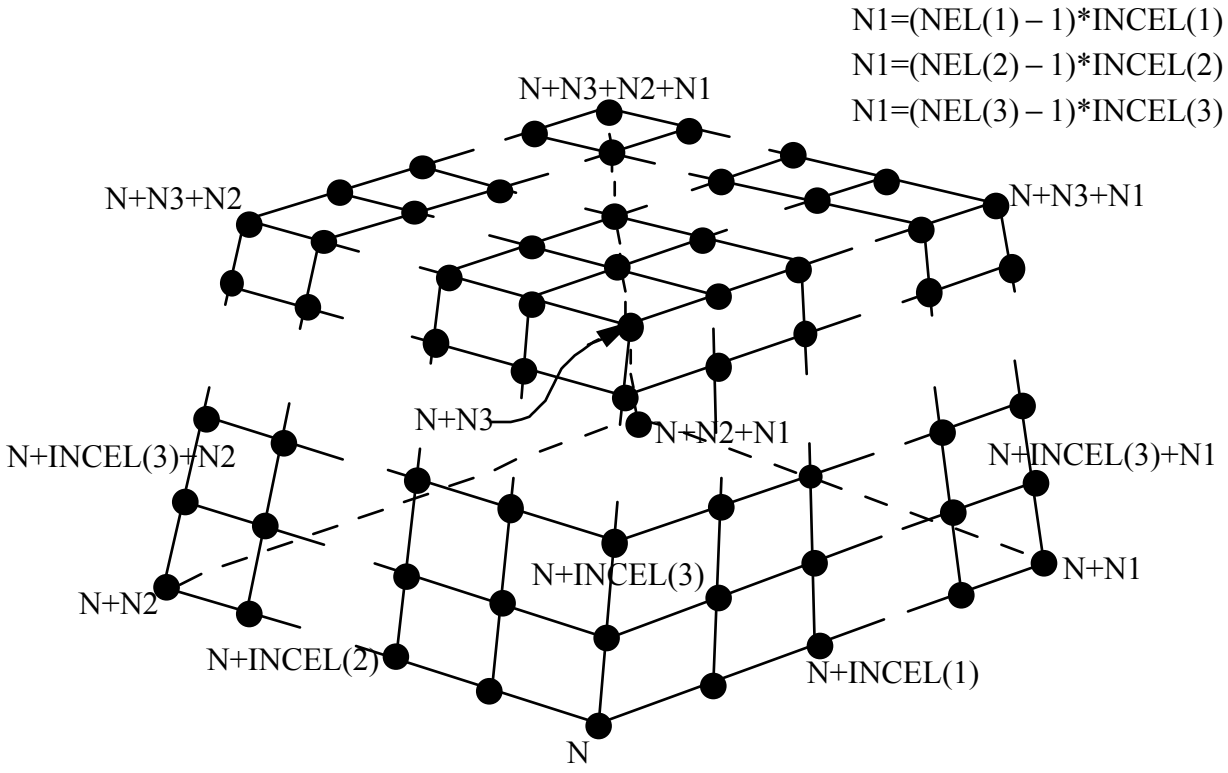


Figure 11.2.2.1 Element Generation

11.2.3 Element Generation Data for 3D Solid Elements (NEN = 8)

Element nodes must be listed in the order shown on Figure 9.2.0.1. See Figure 11.2.3.1 for a schematic representation of the generation scheme.

Note	Variable	Default	Description
	NEL (1)	[0]	Number of elements in direction 1 ≥ 0
	INCEL (1)	[0]	Element number increment for direction; if = 0, set internally to 1
	INC (1)	[0]	Node number increment for direction 1; if = 0, set internally to 1
	NEL (2)	[0]	Number of elements in direction 2 ≥ 0
	INCEL (2)	[0]	Element number increment for direction 2; if = 0, set internally to NEL (1)
	INC (2)	[0]	Node number increment for direction 2; if = 0, set internally to (1+NEL(1))*INC(1)
	NEL (3)	[0]	Number of elements in direction 3 ≥ 0
	INCEL (3)	[0]	Element number increment for direction 3; if = 0, set internally to NEL (1)*NEL (2)
	INC (3)	[0]	Node number increment for direction 3; if = 0, set internally to (1+NEL(2))*INC(2)



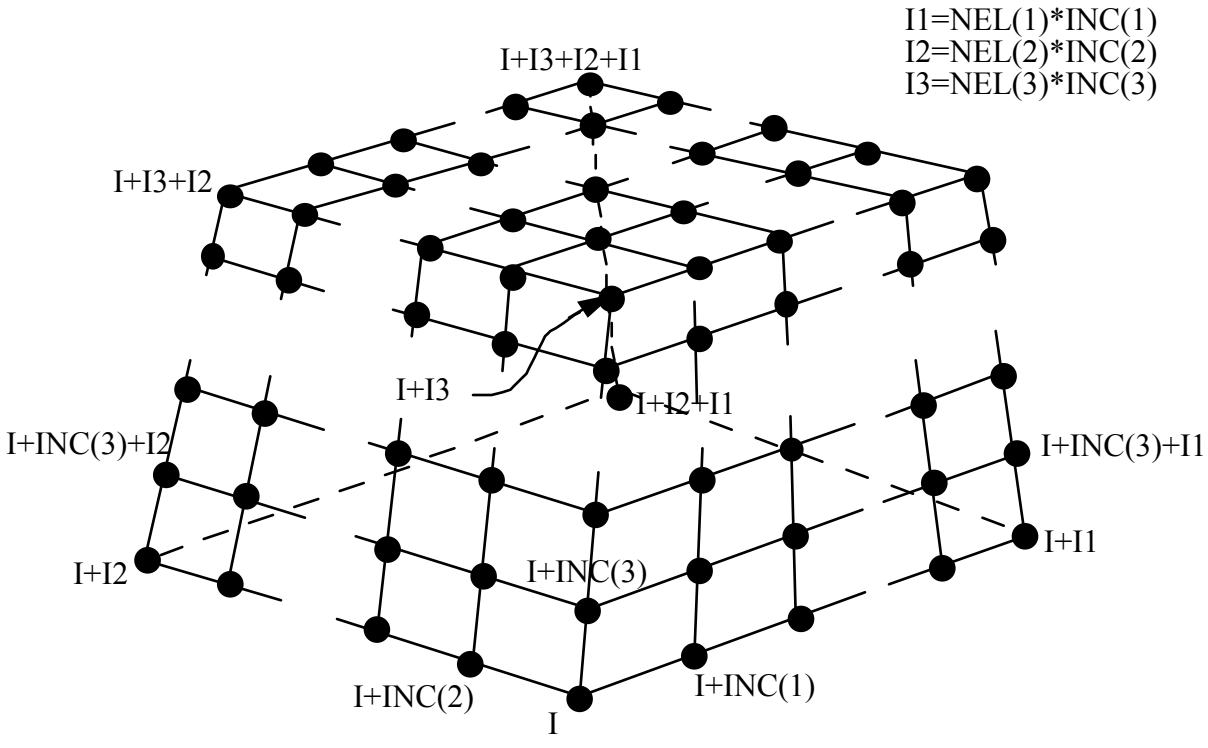


Figure 11.2.3.1 Element Generation

Notes . .

Notes . .

12.0 ANALYSIS OPTIONS

1. The Nonlinear Semi-Discrete Finite Element Equations

The application of the finite element discretization to the governing equation(s) of a field theory generates a matrix system of equations. These equations are either zero-, first- or second-order in the time variable and are generally nonlinear. For example, nonlinear transient finite-element dynamics are characterized by the following second-order semi-discrete balance equation:

$$\mathbf{M}\ddot{\mathbf{d}} + \mathbf{N}\left(\dot{\mathbf{d}}, \mathbf{d}\right) = \mathbf{f} \tag{1}$$

where $\ddot{\mathbf{d}}(t) = \partial^2\mathbf{d}/\partial t^2$ = vector of nodal accelerations; $\dot{\mathbf{d}}(t)$ = vector of nodal velocities, $\mathbf{d}(t)$ = vector of nodal displacements; $\mathbf{M}$ = global mass matrix; $\mathbf{N}$ = global vector-valued function of the displacement $\mathbf{d}$ and the velocity $\dot{\mathbf{d}}$ (when e.g. viscous effects exists); and $\mathbf{f} = \mathbf{f}(t)$ = global vector representing the applied external loads and surface tractions. The global quantities $\mathbf{M}$, $\mathbf{N}$ and $\mathbf{f}$ are assembled from the corresponding finite element matrices and vectors. The vector $\mathbf{N}(\dot{\mathbf{d}}, \mathbf{d})$ is often denoted the "internal force." Note that for particular constitutive models other state variables may also influence $\mathbf{N}$ (for instance, temperature in thermo-continua), but these do not need to be included in the present discussion.

DYNAFLOW is fundamentally structured for transient analysis of nonlinear equation systems, and the solution techniques are presented in that context. Linear balance equations and eigenvalue equations are treated as special cases of the general method. In all cases, any quantities identified as "nodal variables" are to be taken as the approximate values derived from the numerical solution of the finite-element equations.

In describing a unified solution strategy for Eq. 1, it is useful to standardize the notation of the generic second-order system to read:

$$\mathbf{M}\mathbf{a} + \mathbf{N}\left(\mathbf{v}, \mathbf{d}\right) = \mathbf{f} \tag{2}$$

where $\mathbf{a}$, $\mathbf{v}$, and $\mathbf{d}$ are vectors of nodal unknowns with $\mathbf{a} = \ddot{\mathbf{d}}$ and $\mathbf{v} = \dot{\mathbf{d}}$. In the presentation hereafter, the vectors $\mathbf{d}$, $\mathbf{v}$, and $\mathbf{a}$ are referred to as the displacement, velocity and acceleration, respectively. However, their true physical charater is determined by the particular field theory under consideration.

2. Time Integration

The semi-discrete finite-element equation (Eq. 2) is a system of coupled ordinary differential equations which are to be integrated in time. The system is solved by applying a step-by-step integration procedure resulting in a system of nonlinear algebraic equations.

Specifically, the strategy employed is to solve the equations at discrete times t_n (for n = 1, 2, 3...) for the nodal quantities $\mathbf{d}(t_n)$, $\mathbf{v}(t_n)$, and $\mathbf{a}(t_n)$. The procedure starts from user-specified initial conditions:

$$\mathbf{d}(0) = \mathbf{d}_0 \qquad \mathbf{v}(0) = \mathbf{v}_0 \tag{3}$$

and the initial acceleration consistent with Eq. 2 is computed as:

$$\mathbf{a}(0) = \mathbf{a}_0 = \mathbf{M}^{-1}\left[\mathbf{f}_0 - \mathbf{N}(\mathbf{v}_0, \mathbf{d}_0)\right] \tag{4}$$

where $\mathbf{M}^{-1}$ denotes the inverse mass matrix. With the solution fully characterized at time t_0, the objective is to compute the solution at time t_1, then at time t_2 and so on. Numerous algorithms are available for that purpose. DYNAFLOW uses one-step methods to carry out the time integration by assuming auxiliary relationships between the solution vectors at time steps t_n and t_{n+1}. For second-order systems, the Newmark update formulas [see Section 12.3] are used:

$$\mathbf{d}_{n+1} = \mathbf{d}_n + \Delta t\, \mathbf{v}_n + \Delta t^2/2\left[(1-2\beta)\,\mathbf{a}_n + 2\beta\,\mathbf{a}_{n+1}\right] \tag{5}$$

$$\mathbf{v}_{n+1} = \mathbf{v}_n + \Delta t\left[(1-\alpha)\,\mathbf{a}_n + \alpha\,\mathbf{a}_{n+1}\right] \tag{6}$$

$$\mathbf{M}\mathbf{a}_{n+1} + \mathbf{N}(\mathbf{v}_{n+1}, \mathbf{d}_{n+1}) = \mathbf{f}_{n+1} \tag{7}$$

where $\Delta t = t_{n+1} - t_n$; and $\mathbf{d}_n$, $\mathbf{v}_n$, and $\mathbf{a}_n$ are approximations for $\mathbf{d}(t)$, $\mathbf{v}(t)$, and $\mathbf{a}(t)$ respectively, at time t_n, viz.,

$$\mathbf{d}_n \approx \mathbf{d}(t_n) \qquad \mathbf{v}_n \approx \mathbf{v}(t_n) \qquad \mathbf{a}_n \approx \mathbf{a}(t_n) \tag{8}$$

and $\mathbf{f}_{n+1} = \mathbf{f}(t_{n+1})$. The scalars α and β are algorithmic parameters which can be chosen to ensure unconditional stability and second-order accuracy of the time integration in the linear case.

Note that the matrix equation system (Eq.2) can be the composite of several equation types. For example, in solid stress analysis, it might include the equations for stress analysis and thermal analysis. In such cases, each equation type can be assigned its own set of values for the algorithmic parameters α and β. However, the same time steps (t_1, t_2, t_3,...) apply to all equations of the system.

3. Predictor-Multicorrector Strategy

Eq. 7 is a nonlinear system of equations. The solution to this system is obtained by using an iterative strategy which is implemented by means of a predictor-multicorrector scheme

applied at each time step. In this method, a series of corrected solutions are computed starting from an initial or predicted solution for the time step. Each corrected solution is used in the following iteration to compute the next corrected solution. The procedure continues until either a specified solution convergence has been obtained or a pre-specified maximum number of iterations has been executed.

The computational problem at each time step is to determine $\mathbf{d}_{n+1}$, $\mathbf{v}_{n+1}$, and $\mathbf{a}_{n+1}$, given $\mathbf{d}_n$, $\mathbf{v}_n$, $\mathbf{a}_n$, and Δt such that Eqs. 5-7 be satisfied. There are several ways of implementing the recursion relationship that takes the solution from step n to step n + 1. In DYNAFLOW the implementation chosen is to always use the vector of nodal quantities with the highest time derivatives as the primary unknown, viz. for the second-order system of Eqs. 5-7 use the vector of nodal accelerations as the primary variable. Introducing a superscript to index the nonlinear iterations, the Newmark formulas are rewritten as:

$$\mathbf{d}_{n+1}^{(i)} = \mathbf{d}_n + \Delta t\, \mathbf{v}_n + \Delta t^2 / 2\left[(1-2\beta)\, \mathbf{a}_n + 2\beta\, \mathbf{a}_{n+1}^{(i)} \right] \tag{9}$$

$$\mathbf{v}_{n+1}^{(i)} = \mathbf{v}_n + \Delta t\left[(1-\alpha)\, \mathbf{a}_n + \alpha\, \mathbf{a}_{n+1}^{(i)} \right] \tag{10}$$

Substituting these expressions into Eq. 7 result in an equation that may be solved for $\mathbf{a}_{n+1}^{(i)}$ and the acceleration can be determined as:

$$\mathbf{M}\mathbf{a}_{n+1}^{(i)} + \mathbf{N}\left(\mathbf{a}_{n+1}^{(i)}; \mathbf{a}_n, \mathbf{v}_n, \mathbf{d}_n, \Delta t\right) = \mathbf{f}_{n+1} \tag{11}$$

where the unknowns $\mathbf{a}_{n+1}^{(i)}$ are the nodal values of the acceleration at time t_{n+1}. The internal force **N** is a system of nonlinear functionals of the solution $\mathbf{a}_{n+1}^{(i)}$ and of the (known) parameters $\mathbf{a}_n$, $\mathbf{v}_n$, $\mathbf{d}_n$, and Δt. Once $\mathbf{a}_{n+1}^{(i)}$ is determined, Eqs. 9 and 10 serve to determine $\mathbf{v}_{n+1}^{(i)}$ and $\mathbf{d}_{n+1}^{(i)}$.

The solution at step n + 1 is computed by first specifying a starting value for $\mathbf{a}_{n+1}^{(0)}$, the predictor value, which corresponds to an initial guess at the solution at time t_{n+1}, viz.,

$$\mathbf{a}_{n+1}^{(0)} = 0 \tag{12}$$

except at nodes at which the motion is prescribed for which one sets:

$$\mathbf{a}_{n+1}^{(0)} = \ddot{\mathbf{d}}(t_{n+1}) \tag{13}$$

The correctors represent the update to the nodal solution variables during each nonlinear iteration. The corrector formulas are written (from Eqs 9-11) as :

$$\mathbf{a}_{n+1}^{(i+1)} = \mathbf{a}_{n+1}^{(i)} + \Delta\mathbf{a}_{n+1}^{(i)} \tag{14}$$

$$\mathbf{v}_{n+1}^{(i+1)} = \mathrm{v}_{n+1}^{(i)} + \alpha\,\Delta t\,\Delta\mathbf{a}_{n+1}^{(i)} \tag{15}$$

$$\mathbf{d}_{n+1}^{(i+1)} = \mathbf{d}_{n+1}^{(i)} + \beta\,\Delta t\,\Delta\mathbf{a}_{n+1}^{(i)} \tag{16}$$

where $\Delta\mathbf{a}_{n+1}^{(i)}$ = acceleration increments which must be determined. The correctors ensure that every set of iterates $\left(\mathbf{a}_{n+1}^{(i+1)}, \mathbf{v}_{n+1}^{(i+1)}, \mathbf{d}_{n+1}^{(i+1)}\right)$ adhere to the Newmark update formulas. Also note that the correctors of the $(i+1)^{th}$ iteration are computed from those of the i^{th}-iteration and the i^{th}-increment of the acceleration $\Delta\mathbf{a}_{n+1}^{(i)}$.

4. Nonlinear Iterations - Linearization

We write the resulting system of equations to be solved (Eq. 11) as:

$$\mathbf{r}\left(\mathbf{a}_{n+1}; \mathbf{u}_n, \Delta t\right) = \mathbf{f}_{n+1} - \mathbf{n}\left(\mathbf{a}_{n+1}; \mathbf{u}_n, \Delta t\right) = \text{residual} \tag{17}$$

where

$$\mathbf{n}\left(\mathbf{a}_{n+1}; \mathbf{u}_n, \Delta t\right) = \mathbf{M}\mathbf{a}_{n+1} + \mathbf{N}\left(\mathbf{a}_{n+1}; \mathbf{u}_n, \Delta t\right) \tag{18}$$

and $\mathbf{u}_n = \{\mathbf{a}_n, \mathbf{v}_n, \mathbf{d}_n\}$ denotes the collection of known parameters, and $\mathbf{r}$ = residual is a system of nonlinear functionals of the solution $\mathbf{a}_{n+1}$ and of the parameters $\mathbf{u}_n$ and Δt. This system is solved for $\mathbf{a}_{n+1}$ by performing a linearization via a truncated Taylor's series expansion of $\mathbf{r}$ as:

$$\mathbf{r}\left(\mathbf{a}_{n+1}^{(i+1)}; \mathbf{u}_n, \Delta t\right) \approx \mathbf{r}\left(\mathbf{a}_{n+1}^{(i)}; \mathbf{u}_n, \Delta t\right) + \frac{\partial \mathbf{r}}{\partial \mathbf{a}_{n+1}}\left(\mathbf{a}_{n+1}^{(i)}; \mathbf{u}_n, \Delta t\right)\Delta\mathbf{a}_{n+1}^{(i)} = 0 \tag{19}$$

where the solution increment $\Delta\mathbf{a}_{n+1}^{(i)}$ is given by:

$$\mathbf{a}_{n+1}^{(i)} = \mathbf{a}_{n+1}^{(i+1)} - \mathbf{a}_{n+1}^{(i)} \tag{20}$$

$\mathbf{a}_{n+1}^{(i+1)}$ and $\mathbf{a}_{n+1}^{(i)}$ are approximations of $\mathbf{a}_{n+1}$ at iterations i and i + 1, respectively. Denote

$$\mathbf{r}_{n+1}^{(i)} = \mathbf{r}\left(\mathbf{a}_{n+1}^{(i)}; \mathbf{u}_n, \Delta t\right) = \text{residual} \tag{21}$$

$$\mathbf{J}_{n+1}^{(i)} = \frac{\partial \mathbf{r}}{\partial \mathbf{a}_{n+1}}\left(\mathbf{a}_{n+1}^{(i)}; \mathbf{u}_n, \Delta t\right) = \mathbf{J}\left(\mathbf{a}_{n+1}^{(i)}; \mathbf{u}_n, \Delta t\right) = \text{Jacobian matrix} \tag{22}$$

The linear system of equations (Eq.19) to be solved for $\Delta \mathbf{a}_{n+1}^{(i)}$ can be written as:

$$-\mathbf{J}_{n+1}^{(i)}\ \Delta \mathbf{a}_{n+1}^{(i)} = \mathbf{r}_{n+1}^{(i)} \tag{23}$$

and the consistent Jacobian matrix (Eq. 22) is computed from Eqs. 17 and 18 as:

$$\mathbf{J}_{n+1}^{(i)} = -\left.\frac{\partial \mathbf{n}}{\partial \mathbf{a}_{n+1}}\right|_{\left(\mathbf{a}_{n+1}^{(i)}; \mathbf{u}_n, \Delta t\right)} = -\left[\mathbf{M} + \alpha \Delta t \mathbf{C}_{n+1}^{(i)} + \beta \Delta t^2 \mathbf{K}_{n+1}^{(i)}\right] \tag{24}$$

with

$$\mathbf{C}_{n+1}^{(i)} = -\left.\frac{\partial \mathbf{N}(\mathbf{v}, \mathbf{d})}{\partial \mathbf{v}}\right|_{\left(\mathbf{v}=\mathbf{v}_{n+1}^{(i)}; \mathbf{d}=\mathbf{d}_{n+1}^{(i)}\right)} = \text{consistent damping matrix} \tag{25}$$

and

$$\mathbf{K}_{n+1}^{(i)} = -\left.\frac{\partial \mathbf{N}(\mathbf{v}, \mathbf{d})}{\partial \mathbf{d}}\right|_{\left(\mathbf{v}=\mathbf{v}_{n+1}^{(i)}; \mathbf{d}=\mathbf{d}_{n+1}^{(i)}\right)} = \text{consistent stiffness matrix} \tag{26}$$

The linear combination of matrices on the left-hand side of Eq. 23 is called the effective mass matrix $\mathbf{M}^*$ where

$$\mathbf{M}^* = \mathbf{M} + \alpha \Delta t \mathbf{C}_{n+1}^{(i)} + \beta \Delta t^2 \mathbf{K}_{n+1}^{(i)} \tag{27}$$

and Eq. 23 is usually written as:

$$\mathbf{M}^* \Delta \mathbf{a}_{n+1}^{(i)} = \mathbf{r}_{n+1}^{(i)} \tag{28}$$

This equation is solved for $\Delta \mathbf{a}_{n+1}^{(i)}$ and the solution is used in the corrector formulas (Eqs. 14-16) to compute the new iterate $\left(\mathbf{a}_{n+1}^{(i+1)}, \mathbf{v}_{n+1}^{(i+1)}, \mathbf{d}_{n+1}^{(i+1)}\right)$.

Remark: In the linear case, the assembled internal force vector **N(v, d)** in Eq. 2 can be represented as

$$\mathbf{N}(\mathbf{v}, \mathbf{d}) = \mathbf{C}\mathbf{v} + \mathbf{K}\mathbf{d} \tag{29}$$

where the damping and stiffness matrices, **C** and **K** are independent of the velocity and displacement. Then the effective mass matrix in Eq. 28 is given by:

$$\mathbf{M}^* = \mathbf{M} + \alpha\Delta t\mathbf{C} + \beta\Delta t^2\mathbf{K} \tag{30}$$

and needs to be evaluated only when the time step value changes. The expansion in Eq. 19 is then exact, and the solution at step (n+1) is obtained as (from Eq. 28):

$$\Delta\mathbf{a}_{n+1} = \mathbf{M}^{*-1}\,\mathbf{r}_{n+1}^{(0)} \tag{31}$$

and (from Eqs. 14 - 16):

$$\mathbf{a}_{n+1} = \mathbf{a}_{n+1}^{(0)} + \Delta\mathbf{a}_{n+1} \tag{32}$$

$$\mathbf{v}_{n+1} = \mathbf{v}_{n+1}^{(0)} + \alpha\Delta t\Delta\mathbf{a}_{n+1} \tag{33}$$

$$\mathbf{d}_{n+1} = \mathbf{d}_{n+1}^{(0)} + \beta\Delta t^2\Delta\mathbf{a}_{n+1} \tag{34}$$

where the predictors $\mathbf{v}_{n+1}^{(0)}$ and $\mathbf{d}_{n+1}^{(0)}$ are obtained from Eqs. 9, 10, 12 and 13.

Section 12.2 discusses the discrete time integration schemes. The various nonlinear iteration strategies available in DYNAFLOW are described in Section 12.3, and Section 12.4 describes the various linear solvers available to compute the solution to Eq. 28. In Section 12.4, the system of equations in Eq. 28 is simply viewed and solved as the linear algebraic system **Ax** = **b**. Direct and iterative methods are provided. Section 12.5 discusses the various eigensolvers available.

Notes . .

Notes . .

12.1 Define Solution Staggers (Optional)

The procedure allows the use of a multistagger solution strategy in which the full system of coupled equations defined in the problem domain is partitioned into smaller subsystems of equations. Each subsystem is then solved separately assuming that the unknown variables of the other subsystems are temporarily frozen. The equation subsystems are referred to as "staggers," and the approach amounts to a partial decoupling of the full system of coupled equations. Each stagger is assigned its own time-stepping parameters, nonlinear iteration procedure, linear equation solver and/or eigen solver. In each stagger, the active variables are solved while all the other variables belonging to the other staggers are held fixed. In turn, after update of the solution variables of the current stagger, they are held fixed when solving for the variables of the subsequent staggers. The staggers are solved sequentially and repeatedly in a predetermined sequence until they all have converged to a self consistent set of solution variables. The iterative procedure is handled by the Stagger_control command. The procedure is equivalent to solving the coupled system (see e.g. [1-5]). Solution staggers may also be used to selectively solve for the equations defined in various parts of the mesh. Element groups belonging to the same region cannot be assigned to the same solution stagger.

12.1.1 Define Staggers

DEFINE_STAGGER

	DEFINE_STAGGER name = "<string>", etc...
	Set up solution staggers to use for solving the equations defined in the problem.

Note	Variable Name	Type	Default	Description
	Name	string	[none]	Stagger name Name must be enclosed in quotation marks.
(1)	Element_group(s)	string	[none]	Element groups belonging to the stagger. Name(s) must be enclosed in quotation marks
	Variable	list	[all]	Active variable selection for the stagger:
	all			All degrees of freedom
	solid_displacement			Solid displacement
	solid_displacement_and_rotation			Solid displacement and rotation
	fluid_velocity			Fluid velocity
	fluid_velocity_and_pressure			Fluid velocity and pressure
	temperature			Temperature
	potential			Potential
	electric_potential			Electric potential
	pressure			Pressure
	scalar_transport			Scalar transport
	level_set			Level set
	stream_fct			Stream function
	solid_displacement_and_fluid_velocity			Coupled porous continuum (hyperbolic)
	solid_displacement_and_fluid_pressure			Coupled continuum (parabolic)
	solid_displacement_and_temperature			Coupled thermo-solid continuum
	mesh_motion			ALE mesh motion

(cont'd)

(cont'd)

Note	Variable Name	Type	Default	Description
	Activation_time	real	[0.0]	Time at which stagger becomes active
	Deactivation_time	real	[0.0]	Time at which stagger becomes inactive (only active if deactivation_time > activation_time)
	External_loads **on** / off	string	[on]	External loads option
	Nodal_loads **on** / off	string	[on]	Nodal loads option
	Surface_loads **on** / off	string	[on]	Surface loads option
	Heat_loads **on** / off	string	[on]	Heat loads option
	Multi_flow_initialize on / **off**	string	[off]	Multi-flow initialization (viz., $p_2 = p_1$)
(2)	Time_integration	command	[none]	Time integration parameters used in the stagger
	Nonlinear_iterations	command	[none]	Nonlinear iterations used in the stagger
	Linear_solver	command	[none]	Linear solver used in the stagger
	Eigen_solver	command	[none]	Eigen solver used in the stagger

EXAMPLE

```
Define_stagger  /                              # define a solution stagger
     name = "stagger1"  /                      # set name
     element_group(s) =  /                     # include element groups 1 and 2
        "Group1" , "Group2"  /
     time_integration   /                      # time integration parameters, etc...
        equation_type = ... , etc...
```

Notes/

(1) List all element group names or numbers belonging to the solution stagger. Element groups belonging to the same region cannot be assigned to the same solution stagger.

(2) The requested analysis parameters may be entered directly without use of the main commands keywords.

12.1.2 Staggers Control

STAGGER_CONTROL

STAGGER_CONTROL	name = "<string>", etc...

The staggers are solved sequentially and repeatedly in a predetermined sequence until they all have converged to a self consistent set of solution variables. The iterative procedure is handled by the stagger_control command. When solution staggers are used to selectively solve for the equations defined in various parts of the mesh, then an interface between the various physical solution domains will be defined.

Note	Variable Name	Type	Default	Description
	Name	string	[none]	Stagger name Name must be enclosed in quotation marks
	Balance_of_mass on / **off**	string	[off]	Balance of mass option
	Coupling **on** / off	string	[on]	Coupling option
	Interface **on** / off	string	[on]	Interface option
	Solver_name **CG_Mf** / CG_EBE	string	[*]	Solver name
	Max_number_of_iterations	integer	[0]	Maximum number of iterations
	Min_number_of_iterations	integer	[0]	Minimum number of iterations
	Convergence_check **on** / off	list	[on]	Convergence check
	Convergence_tol_sol	real	[1.E-3]	Convergence tolerance for solution ≥ 0.0
	Convergence_tol_mass	real	[0.0]	Convergence tolerance for mass (in %) ≥ 0.0

EXAMPLE

```
Stagger_control /
    max_number_of_iterations = 5 /
    convergence_check = on
```

References / Bibliography

1. Farhat, Charbal, K.C. Park and Yves DuBois-Pelerin, "An Unconditionally Stable Staggered Algorithm for Transient Finite Element Analysis of Coupled Thermoelastic Problems," *Comp. Meth. Appl. Mech. Eng.*, **85**, (1991), 349-365.

2. Felippa, C.A. and K.C. Park, "Staggered transient analysis procedures for coupled mechanics problems: formulation," *Comp. Meth. Appl. Mech. Eng.*, **24**, (1980), 61-111.

3. Park, K.C.,"Partitioned transient analysis procedures for coupled field problems: stability analysis", *J.Appl. Mech.*, **47**, (1980), 370-376.

4. Park, K.C. and C.A. Felippa,"Partitioned transient analysis procedures for coupled field problems: accuracy analysis", *J. Appl. Mech.*, **47**, (1980), 919-926.

5. Prevost, J. H., "Partitioned Solution Procedure for Simultaneous Integration of Coupled-Field Problems," *Comm. Num. Meth. Eng.*, **13**, No. 4, (1997), pp. 239-247.

Notes . .

12.2 Time-Integration Parameters

TIME_INTEGRATION

TIME_INTEGRATION	integration_type = etc...

Specify the parameters for the time stepping procedure to use for all equations in all subdomains/element groups belonging to any given solution stagger.

Note	Variable Name	Type	Default	Description
(1)	Integration_type **implicit** / explicit	list	[*]	Implicit/explicit time integration type
(2)	Equation_type **elliptic** parabolic hyperbolic	list	[*]	Equation type (see Chapter 9 for details) Elliptic Boundary Value problem Parabolic Initial Boundary Value problem Hyperbolic Initial Boundary Value problem
	Analysis_type **direct** modal spectral Runge_Kutta_Cash_Karp Runge_Kutta_Bulirsh_Stoer	list	[*]	Analysis type Direct one-step time integration Modal integration Spectral integration Runge_Kutta fifth-order Cash_Karp integration with adaptive stepsize control Runge_Kutta integration with Bulirsh_Stoer steps and adaptive stepsize control
	Alpha	real	[1.0]	Algorithmic parameter $\alpha \geq 0.0$ = 1.0 for Elliptic BVP ≥ 0.0 for Parabolic Initial BVP ≥ 0.5 for Hyperbolic Initial BVP
(3)	Beta	real	[0.0]	Algorithmic parameter $\beta \geq 0.0$
• *Modal and Spectral Analysis Options*				
	Number_of_modes	integer	[0]	Number of modes
	Modal_damping_ratio	real	[0.0]	Modal damping ratio
• *Spectral Analysis Option*				
	Spectrum_load_time	integer	[0]	Spectrum load-time function number ≥ 1

EXAMPLE

```
Time_integration /
   Integration_type = implicit /      # implicit time integration
   Equation_type = hyperbolic /       # hyperbolic initial BVP
   Alpha = 0.5,   Beta = 0.25         # Select Trapezoidal rule
```

Notes/
(1) Explicit time integration is performed using a diagonal mass matrix.

Notes / (cont'd)

(2) The application of the finite element discretization to the governing balance equation(s) of a field theory generates a matrix system of equations. These equations are either zero-, first- or second order in the time variable, and are referred to in the following as *elliptic*, *parabolic* and *hyperbolic*, respectively. One-step algorithms are used to integrate the finite element semidiscrete equations of motion as follows (for simplicity in the presentation linear systems are used in the following):

A. *Hyperbolic and Parabolic-Hyperbolic Initial Boundary Value Problems:* the Newmark [3] family of finite difference time stepping algorithms is used which consist of the following equations:

$$\mathbf{M}\mathbf{a}_{n+1} + \mathbf{C}\mathbf{v}_{n+1} + \mathbf{K}\mathbf{d}_{n+1} = \mathbf{f}_{n+1}$$

$$\mathbf{d}_{n+1} = \mathbf{d}_n + \Delta t \mathbf{v}_n + \Delta t^2/2 \left[(1-2\beta)\mathbf{a}_n + 2\beta\,\mathbf{a}_{n+1}\right]$$

$$\mathbf{v}_{n+1} = \mathbf{v}_n + \Delta t \left[(1-\alpha)\mathbf{a}_n + \alpha\,\mathbf{a}_{n+1}\right]$$

where α is a parameter taken to be in the interval $\alpha \in$ [1/2, 3/2] and $\beta \geq 0$. Unconditional stability requires that β be taken to be in the interval $\beta \in [\alpha/2, 1]$. Maximal high-frequency numerical dissipation is provided by selecting [1] $\beta = (\alpha + 1/2)^2 / 4$ for a given $\alpha \geq 1/2$. Some well-known integrators are identified as follows:

$\alpha = 1/2,\ \beta = 0$: Explicit central difference
$\alpha = 1/2,\ \beta = 1/4$: Trapezoidal

A particularly convenient form of **C** is the Rayleigh damping matrix:

$$\mathbf{C} = a_0\mathbf{M} + a_1\mathbf{K}$$

where a_0 and a_1 are parameters (see e.g., Section 10.1) referred to as mass and stiffness damping, respectively. Then, the resulting viscous damping can be computed as:

$$\xi_i = \frac{1}{2}\left(\frac{a_0}{\omega_i} + a_1\omega_i\right)$$

for each modal frequency i (i = 1, neq) see Fig. 12.2.1. The parameters a_0 and a_1 may be selected to produce desired damping characteristics (e.g., by adjusting a_0 and a_1 for two eigenfrequencies).

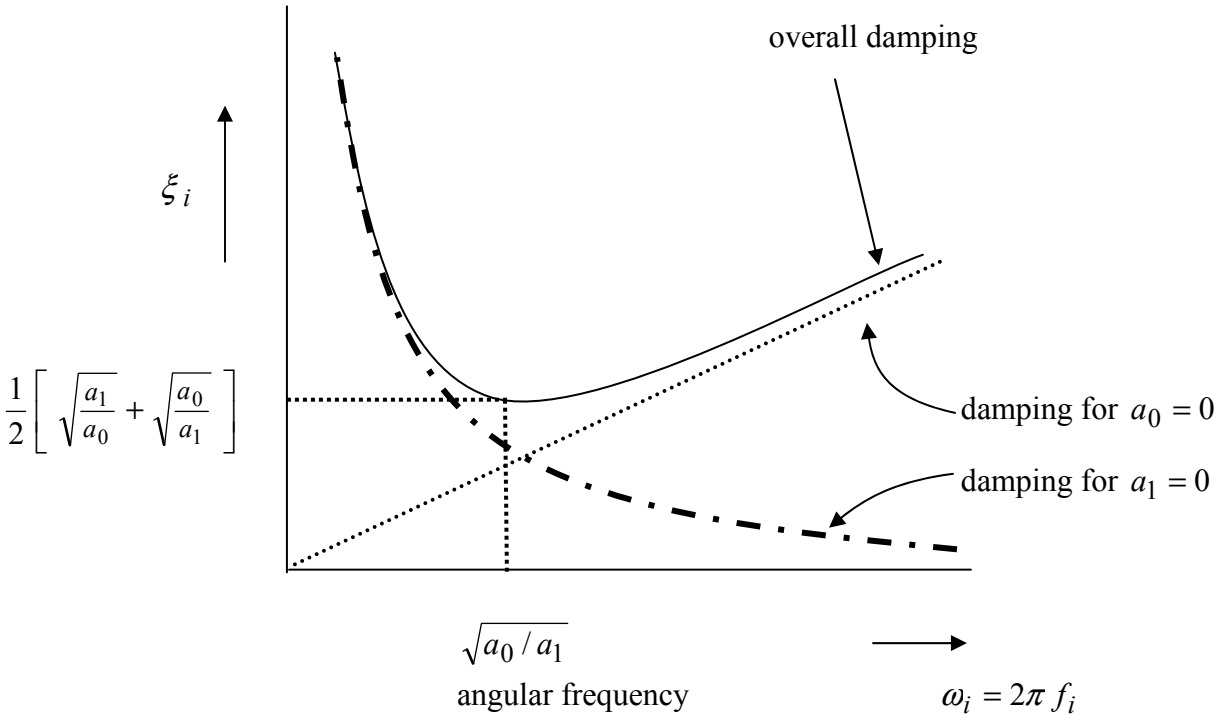


Fig. 12.2.1 Effect of Viscous Damping

As a result of the numerical integration, artificial damping and period distortion are introduced. The following results have been obtained:

$$\bar{\xi} = \xi + \frac{1}{2}\left(\alpha - \frac{1}{2}\right)\Omega + \mathrm{O}\left(\Omega^2\right)$$

$$\frac{\varpi - \omega}{\omega} = \mathrm{O}\left(\Omega^2\right) \qquad \Omega = \omega\Delta\mathrm{t}$$

Where $\bar{\xi}$ = algorithmic damping ratio, ϖ = algorithmic frequency. Note that first-order errors resulting from $\alpha \neq 1/2$ manifest themselves only in the form of excess numerical dissipation, and not in period discrepancies.

B. *Parabolic Initial Boundary Value Problems*: the generalized trapezoidal family of finite difference time stepping algorithms is used which consist of the following equations:

$$\mathbf{M}\mathbf{v}_{n+1} + \mathbf{K}\mathbf{d}_{n+1} = \mathbf{f}_{n+1}$$

$$\mathbf{d}_{n+1} = \mathbf{d}_n + \Delta t\, \mathbf{v}_{n+\alpha}$$

$$\mathbf{v}_{n+\alpha} = (1-\alpha)\mathbf{v}_n + \alpha\, \mathbf{v}_{n+1}$$

where α is a parameter taken to be in the interval $\alpha \in [0, 1]$. Unconditional stability requires that □ be taken to be in the interval $\alpha \in [1/2, 1]$. Maximal high-frequency numerical dissipation is provided by selecting $\alpha = 1$. Some well-known integrators are identified as follows:

$\alpha = 0$: Explicit forward Euler
$\alpha = 1/2$: Crank Nicolson/midpoint rule
$\alpha = 1$: Implicit backward Euler

C. *Elliptic Boundary Value Problems:* a backward finite difference (backward Euler) time stepping algorithms is used which consist of the following equations:

$$\mathbf{K}\mathbf{d}_{n+1} = \mathbf{f}_{n+1}$$

$$\mathbf{d}_{n+1} = \mathbf{d}_n + \Delta t\, \mathbf{v}_{n+1}$$

(3) For *implicit* time integration of hyperbolic IBVP, if $\beta = 0$, set internally to:

$$\beta = (\alpha + 1/2)^2 / 4$$

References / Bibliography

1. Hilber, H.M., "Analysis and Design of Numerical Integration Methods in Structural Dynamics," *Rep. No. EERC 76-29*, Earthquake Engineering Research Center, University of California, Berkeley, CA, (1976).

2. Hughes, T.J.R., Pister, K.S., and Taylor, R.L., "Implicit-Explicit Finite Elements in Nonlinear Transient Analysis," *Comp. Meth. Appl. Mech. Eng.*, **17**/**18**, (1979), 159-182.

3. Newmark, N.M.,"A Method of Computation for Structural Dynamics," *J. Eng. Mech. Div.*, ASCE, **85**, EM3, (1959), 67-94.

Notes . .

Notes . .

12.3 Nonlinear Iteration Requests

NONLINEAR_ITERATIONS

	NONLINEAR_ITERATIONS	iteration_type = list , etc...

Specify the parameters for nonlinear iterations to use for all equations in all sub-domains/element groups belonging to any given solution stagger.

Note	Variable Name	Type	Default	Description
(1)	Iteration_type **modified_Newton_Raphson** Newton_Raphson quasi_Newton_BFGS quasi_Newton_Strang_BFGS quasi_Newton_Broyden Linear	list	[*]	Nonlinear Iteration procedure: Modified Newton-Raphson iterations Newton-Raphson iterations Quasi-Newton iterations (BFGS update) Quasi-Newton iterations (BFGS product) Quasi-Newton iterations (Broyden update) Linear iterations
(2)	Max_number_of_iterations	integer	[0]	Maximum number of iterations
	Min_number_of_iterations	integer	[0]	Minimum number of iterations.
	Convergence_check **on** / off	list	[on]	Convergence check
	Convergence_tol_rhs	real	[1.E-3]	Convergence tolerance for residual ≥ 0.0
	Convergence_tol_sol	real	[1.E-3]	Convergence tolerance for solution ≥ 0.0
(7)	Convergence_norm **L2_norm** infinite_norm other	list	[*]	Convergence norm
	Relative_tolerance	real	[1.E-3]	Relative tolerance Rtol
	Absolute_tolerance	real	[0.0]	Absolute tolerance Atol
(3)	Jacobian_matrix on / **off**	list	[off]	Jacobian matrix

(cont'd)

(cont'd)

Note	Variable Name	Type	Default	Description
(3)	Jacobian_derivative **backward** forward central	list	(*)	Jacobian derivative option
(4)	Reform_lhs_step	integer	[0]	Step number at which left-hand side (Stiffness) reforms are to be initiated.
(4)	Reform_lhs_freq	integer	[0]	Left-hand side (Stiffness) reform frequency
(4)	Reform_lhs_iter	integer	[0]	Left-hand side (Stiffness) reform frequency during iterations
	Max_update_vectors	integer	[0]	Maximum number of updating vectors (only applicable to Quasi-Newton iterations)
(5)	Line_search_type Strang line_minimization backtracking backtrack_line_minimization backtrack_Strang	list	[none]	Line search requests
	Line_search_freq	integer	[1]	Line search frequency (Default: every time step)
(6)	Scaling on / **off**	list	[off]	Scaling

EXAMPLE

```
Nonlinear_Iterations  /
Iteration_type = Quasi_Newton_BFGS /   # Quasi-Newton BFGS iterations request
Max_number_of_iterations = 10 /        # up to 10 iterations per time step
Convergence_tol_rhs = 1.E-6 /          # specified convergence tolerance for residual
Convergence_tol_sol = 1.E-6 /          # specified convergence tolerance for solution
Reform_lhs_freq = 2 /                  # reform lhs every 2 time step
Max_update_vectors = 5 /               # maximum number of BFGS vectors
Line_search_type = Strang /            # perform line search with Strang's algorithm
Line_search_freq = 1                   # line search every time step
```

Notes /

(1) Given the solution $\mathbf{u}_n$ at time t_n and a time increment δt, the object is to find the solution vector $\mathbf{u}$ at time $t_{n+1} = t_n + \delta t$, which satisfies equilibrium. We write the resulting system of equations to be solved at each time/load step as:

$$\mathbf{r}(\mathbf{u}; \mathbf{u}_n, \delta t) = \mathbf{f}_{n+1} - \mathbf{n}(\mathbf{u}; \mathbf{u}_n, \delta t) = 0 \tag{1}$$

where the unknowns $\mathbf{u}$ are the nodal values at time t_{n+1}, and the equations express a balance between external ($\mathbf{f}_{n+1}$) and internal ($\mathbf{n}(\mathbf{u})$) forces. Both $\mathbf{r}$ and $\mathbf{u}$ are vectors, and the residual $\mathbf{r}$ is a system of nonlinear functionals of the solution $\mathbf{u}$ and of the parameters $\mathbf{u}_n$ and δt. This system is solved for $\mathbf{u}$ by performing a linearization via a truncated Taylor's series expansion of $\mathbf{r}$ as:

$$\mathbf{r}(\mathbf{u}_n + \delta\mathbf{u}; \mathbf{u}_n, \delta t) \approx \mathbf{r}(\mathbf{u}_n; \mathbf{u}_n, \delta t) + \frac{\partial \mathbf{r}}{\partial \mathbf{u}}(\mathbf{u}_n, \delta t) \cdot \delta\mathbf{u} = 0 \tag{2}$$

where $\delta\mathbf{u} = \mathbf{u} - \mathbf{u}_n$ and

$$\mathbf{r}(\mathbf{u}; \mathbf{u}_n, \delta t) = \mathbf{f}_{n+1} - \mathbf{n}(\mathbf{u}_n) = \mathbf{r}(\mathbf{u}_n) + \delta\mathbf{f} \approx \delta\mathbf{f} \tag{3}$$

with $\delta\mathbf{f} = \mathbf{f}_{n+1} - \mathbf{f}_n$ and $\mathbf{r}(\mathbf{u}_n) = \mathbf{f}_n - \mathbf{n}(\mathbf{u}_n) \approx 0$. The matrix of the first derivatives $\mathbf{J} = \frac{\partial \mathbf{r}}{\partial \mathbf{u}}$ in Eq. 2 is the consistent (tangent) Jacobian matrix (e.g., $\mathbf{J} = -\left(\mathbf{M} + \alpha\Delta t\, \mathbf{C} + \beta\Delta t^2 \mathbf{K}\right)$ for second-order systems). The linear system of equations to be solved (Eq. 2) can be written as:

$$\mathbf{J} \cdot \delta\mathbf{u} = -\mathbf{r}; \qquad \delta\mathbf{u} = -\mathbf{J}^{-1} \cdot \mathbf{r} \tag{4}$$

- If the system of equations (Eq. 1) is linear, the expansion in Eq. 2 is exact, and the solution at step n+1 is obtained as:

$$\mathbf{u}_{n+1} = \mathbf{u} = \mathbf{u}_n + \delta\mathbf{u}; \qquad \mathbf{r}(\mathbf{u}_n + \delta\mathbf{u}; \mathbf{u}_n, \delta t) = 0 \tag{5}$$

- If the system of equations (Eq. 1) is nonlinear, it is solved for $\mathbf{u}$ by performing a succession of linearizations, leading to Newton's algorithm. This is done via a truncated Taylor series expansion for $\mathbf{r}$:

$$\mathbf{r}\left(\mathbf{u}^{(i+1)}; \mathbf{u}_n, \delta t\right) \approx \mathbf{r}\left(\mathbf{u}^{(i)}; \mathbf{u}_n, \delta t\right) + \frac{\partial \mathbf{r}}{\partial \mathbf{u}}\left(\mathbf{u}^{(i)}; \mathbf{u}_n, \delta t\right) \cdot \mathbf{p}^{(i)} = 0 \tag{6}$$

where the solution increment $\mathbf{p}^{(i)}$ is given by:

$$\mathbf{p}^{(i)} = \mathbf{u}^{(i+1)} - \mathbf{u}^{(i)} \tag{7}$$

(Notes / (cont'd)

$\mathbf{u}^{(i+1)}$ and $\mathbf{u}^{(i)}$ are approximations of $\mathbf{u}$ at iterations i and i+1, respectively. Denote

$$\mathbf{r}^{(i)} = \mathbf{r}\left(\mathbf{u}^{(i)}; \mathbf{u}_n, \delta t\right) = \text{residual} \tag{8}$$

$$\mathbf{J}^{(i)} = \frac{\partial \mathbf{r}}{\partial \mathbf{u}}\left(\mathbf{u}^{(i)}; \mathbf{u}_n, \delta t\right) = \text{consistent Jacobian} \tag{9}$$

The linear system of equations (Eq.6) can be written as:

$$\mathbf{J}^{(i)} \cdot \mathbf{p}^{(i)} = -\mathbf{r}^{(i)} \tag{10}$$

and the new solution is obtained as:

$$\mathbf{u}^{(i+1)} = \mathbf{u}^{(i)} + \mathbf{p}^{(i)} \tag{11}$$

Use of the consistent Jacobian $\mathbf{J}^{(i)}$ as the left-hand side yields in general the fastest convergence provided the initial guess is within the radius of convergence of the Newton's algorithm. However, it is often more convenient and computationally more expedient to use an approximation to the Jacobian matrix. In general the consistent Jacobian $\mathbf{J}^{(i)}$ in Eq. 10 is thus replaced by a *Jacobian-like* matrix $\boldsymbol{J}^{(i)}$, and the linear systems of equations that need to be solved at each iteration reads:

$$\boldsymbol{J}^{(i)} \cdot \mathbf{p}^{(i)} = -\mathbf{r}^{(i)}; \qquad \boldsymbol{J}^{(i)} \approx \mathbf{J}^{(i)} \tag{12}$$

An effective representation of the tangent (Jacobian) matrix is required in order to accurately capture the solution of the nonlinear problem. This can of course be accomplished through a strict Newton consistent tangent method, but this entails unattractive computational costs since each iteration requires the computation and assembly of the tangent matrix at the current state followed by solution of the linear system of Eq. 10. Further, it requires that the analytic form of the consistent Jacobian derivative be available (the cost of a finite-difference determination of the Jacobian is deemed to be prohibitive). However, certain amounts of tangent information can be inferred from the solution and residual iterates computed within any Newton-like method. Quasi-Newton updates [2,5], also referred to as secant updates, exploit just such information.

Various Newton-like iterative procedures and quasi-Newton updates are available as listed.

- *Newton-Raphson Iterations*: the tangent matrix is recomputed (and assembled if applicable) at every time step and at every iteration. If the consistent Jacobian derivative is returned by all the subdomains/element groups belonging to the given solution stagger, then the method reduces to a true Newton algorithm. Otherwise, a Newton-like iterative algorithm is obtained.

(Notes / (cont'd)

- *Modified Newton-Raphson Iterations*: the iterations are performed with the original matrix available at the beginning of the time/load step. The matrix may be updated by requesting a reform at the beginning of the time/load step (Reform_lhs_freq command).

- *Quasi-Newton Iterations:* both BFGS and Broyden rank two updates are available. The number of updating vectors to be stored is controlled by the command: Max_update_vectors. The number of updating vectors must be less or equal to the number of requested maximum number of iterations.

Two implementations of the BFGS rank two update are available. The Strang_BFGS implementation [12] is restricted to symmetric positive definite matrices, whereas the other BFGS implementation follows its original (and equivalent) form but avoids computing square roots which could be troublesome in the indefinite case. The Broyden update [3] is the most general and is not restricted to symmetric matrices. However, if applied to symmetric matrices, it may not preserve the original symmetry of the system of equations.

Quasi-Newton Methods

Newton's method as described above is quite powerful, but it still has several disadvantages. One major drawback is that the Jacobian matrix is needed. In many cases analytic derivatives are not available, and the cost of a finite-difference approximation of the Jacobian is deemed to be prohibitive. Quasi-Newton methods provide cheap approximations to the Jacobian. These methods are often call *secant methods,* since they reduce to the secant method in one-dimension. Let us denote the approximate Jacobian by $\boldsymbol{J}$. Then the ith quasi-Newton step $\delta\mathbf{u}^{(i)}$ is the solution of

$$J^{(i)} \cdot \delta\mathbf{u}^{(i)} = -\mathbf{r}^{(i)} \tag{13}$$

where $\delta\mathbf{u}^{(i)} = \mathbf{u}^{(i+1)} - \mathbf{u}^{(i)} = \mathbf{p}^{(i)}$ (Eq. 11). The quasi-Newton or secant condition is that $\boldsymbol{J}^{(i+1)}$ satisfy the following equation:

$$\boldsymbol{J}^{(i+1)} \cdot \delta\mathbf{u}^{(i)} = \delta\mathbf{r}^{(i)} \tag{14}$$

where $\delta\mathbf{r}^{(i)} = \mathbf{r}^{(i+1)} - \mathbf{r}^{(i)}$ This is the generalization to multidimension of the one-dimensional secant approximation to the derivative $\partial\mathbf{r}/\partial\mathbf{x}$. However, Eq. 14 does not determine $\boldsymbol{J}^{(i+1)}$ uniquely in more than one dimension. Many different auxiliary conditions to nail down $\boldsymbol{J}^{(i+1)}$ have been explored, but one of the best-performing algorithms in practice results from Broyden's formula [3]. Another update formula was devised by Broyden, Fletcher, Goldfarb and Shanno and is referred to as the BFGS update (see e.g. [5]).

(Notes / (cont'd)

Broyden Update Formula: The formula is based on the idea of getting $\boldsymbol{J}^{(i+1)}$ by making the least change to $\boldsymbol{J}^{(i)}$ consistent with the secant equation (Eq. 14). Broyden showed that the resulting formula is :

$$\boldsymbol{J}^{(i+1)} = \boldsymbol{J}^{(i)} + \frac{\left(\delta\mathbf{r}^{(i)} - \boldsymbol{J}^{(i)} \cdot \delta\mathbf{u}^{(i)}\right) \otimes \delta\mathbf{u}^{(i)}}{\delta\mathbf{u}^{(i)} \cdot \delta\mathbf{u}^{(i)}} \tag{15}$$

The formula can be expressed in terms of the inverse by using the Sherman-Morrison formula [14, 15] to invert Eq. 15 analytically as:

$$\left[J^{(i+1)}\right]^{-1} = \left[J^{(i)}\right]^{-1} + \frac{\left(\delta\mathbf{u}^{(i)} - [\boldsymbol{J}^{(i)}]^{-1} \cdot \delta\mathbf{r}^{(i)}\right) \otimes \delta\mathbf{u}^{(i)} \cdot [\boldsymbol{J}^{(i)}]^{-1}}{\delta\mathbf{u}^{(i)} \cdot [\boldsymbol{J}^{(i)}]^{-1} \cdot \delta\mathbf{r}^{(i)}} \tag{16}$$

BFGS Formula: It is most conveniently written directly in terms of $[\mathbf{J}^{(i)}]^{-1}$ rather than $\mathbf{J}^{(i)}$, and has the form [12]:

$$\left[\boldsymbol{J}^{(i+1)}\right]^{-1} = \left(\mathbf{I} + \mathbf{w}^{(i)} \otimes \mathbf{v}^{(i)}\right)\left[J^{(i)}\right]^{-1}\left(\mathbf{I} + \mathbf{v}^{(i)} \otimes \mathbf{w}^{(i)}\right) \tag{17}$$

where $\mathbf{v}^{(i)}$ and $\mathbf{w}^{(i)}$ are update vectors. Note that the product structure of the BFGS update formula implies that $[\boldsymbol{J}^{(i+1)}]^{-1}$ inherits the symmetry (if applicable) of $[\boldsymbol{J}^{(i)}]^{-1}$ since the two new factors are transposes of one another. Also, if $[\boldsymbol{J}^{(i)}]^{-1}$ is positive-definite and $\mathbf{v}^{(i)} \cdot \mathbf{w}^{(i)} \neq 1$, then $[\boldsymbol{J}^{(i+1)}]^{-1}$ is also positive-definite. Multiplying out the right-hand side of Eq. 17 shows that $[\boldsymbol{J}^{(i+1)}]^{-1}$ is the sum of $[\boldsymbol{J}^{(i)}]^{-1}$ and two rank-one matrices. For this reason, the BFGS update is called a "rank-two update." Formula for the explicit definitions for the update vectors $\mathbf{v}^{(i)}$ and $\mathbf{w}^{(i)}$ can be found in [12] as:

$$\mathbf{w}^{(i)} \equiv \frac{\delta\mathbf{u}^{(i)}}{\delta\mathbf{u}^{(i)} \cdot \delta\mathbf{r}^{(i)}} \tag{18}$$

$$\mathbf{v}^{(i)} \equiv -\delta\mathbf{r}^{(i)} + \alpha^{(i)}\mathbf{r}^{(i)} \tag{19}$$

where

$$\alpha^{(i)} \equiv \sqrt{\frac{-\mathrm{s}^{(i)}\delta\mathbf{r}^{(i)} \cdot \delta\mathbf{u}^{(i)}}{\mathbf{r}^{(i)} \cdot \delta\mathbf{u}^{(i)}}} \tag{20}$$

(Notes / (cont'd)

Note that the line search parameter $s^{(i)}$ enters the definition of $\alpha^{(i)}$. This factor arises when a line search procedure is incorporated within the nonlinear solution strategy [see Note 4]. Otherwise, $s^{(i)} = 1.0$ by default.

The above definition of the BFGS update formula is referred to as the Strang_BFGS and is restricted to symmetric positive definite matrices. For problems in which the matrix is not positive definite, another implementation of the BFGS update is available which follows its original (and equivalent) form but avoids computing the square root in Eq. 20 which can be troublesome in the indefinite case. The update formula in that case is defined as:

$$\left[J^{(i+1)}\right]^{-1} = \left(\mathbf{I} - \beta^{(i)}\delta\mathbf{u}^{(i)} \otimes \delta\mathbf{r}^{(i)}\right)\left[J^{(i)}\right]^{-1}\left(\mathbf{I} - \beta^{(i)}\delta\mathbf{r}^{(i)} \otimes \delta\mathbf{u}^{(i)}\right) + \beta^{(i)}\delta\mathbf{u}^{(i)} \otimes \delta\mathbf{u}^{(i)} \quad (21)$$

where

$$\beta^{(i)} \equiv \frac{1}{\delta\mathbf{r}^{(i)} \cdot \delta\mathbf{u}^{(i)}} \quad (22)$$

Implementation Issues: The use of inverse matrices in the update formula (Eqs. 16, 17, and 21) is purely formal. The actual computation of the inverse is never carried out. Only means of replicating its action upon any vector is required. With $\boldsymbol{J}^{(i)}$ unaltered, the only additional storage required consists of two sets of update vectors (e.g. $\{\mathbf{v}^{(i)}, \mathbf{w}^{(i)}\}_{i=1,\,k}$ for the Strang_BFGS). Typically, a strategy is adopted which limits the maximum number of updates k to some practical value, e.g. $k_{max} = 10$, and is a user-specified parameter value Max_update_vectors, restricted to be less than 20. The default is min (max_number_iter, 20).

Attaining this limit is viewed as a signal to the nonlinear solution driver that the matrix should be reformed. Depending upon the situation, the update procedure is either reinitialized and the set of Quasi-Newton update vectors overwritten as new pairs are generated, or the new update vectors are accumulated over the old ones by sliding down the set of vectors loosing the oldest ones.

(2) Up to the prescribed maximum number of iterations will be performed, unless convergence is detected earlier, i.e. if both:

$$\left\|\mathbf{r}\left(\mathbf{u}^{(i)}\right)\right\| \le \text{tol\_rhs}\left\|\mathbf{r}\left(\mathbf{u}^{(0)}\right)\right\| \text{ .and. } \left\|\mathbf{u}^{(i)} - \mathbf{u}^{(i-1)}\right\| q \le \text{tol\_sol}\left\|\mathbf{u}^{(1)} - \mathbf{u}^{(0)}\right\|(1-q)$$

(Notes / (cont'd)

where $\mathbf{r}(\mathbf{u}^{(i)}) = \mathbf{f} - \mathbf{n}(\mathbf{u}^{(i)})$ = residual, $\mathbf{u}^{(i)}$ = solution at iteration i, $\| \cdot \|$ = denotes the Euclidean norm and q is a contraction factor:

$$q = \max\left(\left\| \mathbf{u}^{(i)} - \mathbf{u}^{(i-1)} \right\| \Big/ \left\| \mathbf{u}^{(i-1)} - \mathbf{u}^{(i-2)} \right\|, \left\| \mathbf{u}^{(i-1)} - \mathbf{u}^{(i-2)} \right\| \Big/ \left\| \mathbf{u}^{(i-2)} - \mathbf{u}^{(i-3)} \right\| \right)$$

where tol_rhs and tol_sol are user-specified convergence tolerances for the residual and the solution, respectively (see Command).

(3) Numerical differentiation is used to compute the consistent Jacobian effective stiffness matrix.

(4) In order to speed up convergence, reform of the effective stiffness matrix may be requested to be performed with a certain time step frequency. Note that by default, for Newton-Raphson iterations a reform is performed at every time/load step for every iteration.

(5) Line searches may be used to speed up convergence in nonlinear analyses. Line search iterations involve a single free parameter s which is determined by means of a scalar-valued objective function or metric. The strategy is then to move to a new point $\mathbf{u}^{(i)}$ along the direction of the Newton-like step $\mathbf{p}^{(i)}$ (not necessarily all the way) as:

$$\mathbf{u}^{(i+1)} = \mathbf{u}^{(i)} + s^{(i+1)}\mathbf{p}^{(i+1)}$$

such that the scalar-valued function has "decreased" sufficiently. For simplicity in the notation, we drop subscript (step number) and superscript (iteration number) in the following.

Line Searches with Strang Option*:* The function employed in this case [12] is the inner product of the solution increment with the recomputed residual, viz.,

$$g(s) = \mathbf{p}^{T}\mathbf{r}(\mathbf{u} + s\mathbf{p})$$

The goal is then to find the root of:

$$g(s) = 0$$

and a method of successive approximations is used to obtain the solution to this equation. Trial values of s are selected based upon an interpolation/extrapolation procedure that Dahlquist and Bjork [4] refer to as the "Illinois Algorithm." Additional logic has been added to increase the robustness of the nonlinear iterations. Note that each evaluation of g(s)

(Notes / (cont'd)

requires the computation of a new residual corresponding to a new trial update. Although the line search iteration is formally motivated by finding the root of g(s) = 0, in practice seeking this condition is too computationally intensive. More typically, a relaxed convergence criterion is used:

$$|g(s)| < \varepsilon |g(s) = 0|$$

where the convergence tolerance may be of the order of 0.5. In our implementation the value $\varepsilon = 0.5$ has been adopted. Once the line search iteration has converged, the last trial update of the solution variables is accepted as the update for the ongoing Newton-like iteration.

In order to increase the robustness and/or efficiency of the Newton-like algorithm, a control over the step length **p** may be used during the iterations. Several techniques for unconstrained optimization have been studied in recent years; see Ref.[6,11] for an overview of these techniques. The idea is to determine the step length s**p** by performing the minimization of a univariate function representing a measure of the residual. The function is defined as follows. Let

$$f(\mathbf{u}) = \frac{1}{2}\mathbf{r}^{T}(\mathbf{u})\mathbf{r}(\mathbf{u}) = \frac{1}{2} < \mathbf{r}, \mathbf{r} >$$

and

$$f(s) = f(\mathbf{u} + s\mathbf{p}); \qquad f'(s) = \nabla f^{T}\mathbf{p}$$

where **u** is the current solution, **p** is the step given by the Newton-like algorithm, and s is the line search parameter. We have trivially:

$$f(0) = f(\mathbf{u}); \qquad f(1) = f(\mathbf{u} + \mathbf{p})$$

Moreover the first derivative of f at $s = 0$ is:

$$f'(0) = \nabla f^{T}(\mathbf{u})\mathbf{p} = \mathbf{r}^{T}(\mathbf{u})\mathbf{J}(\mathbf{u})\mathbf{p}$$

where $\mathbf{J}(\mathbf{u})$ is the Jacobian of $\mathbf{r}(\mathbf{u})$. The matrix-vector product $\mathbf{J}(\mathbf{u})\mathbf{p}$ is computed via the central finite difference approximation:

$$\mathbf{J}(\mathbf{u})\mathbf{p} = \left(\mathbf{r}(\mathbf{u} + \varepsilon\mathbf{p}) - \mathbf{r}(\mathbf{u} - \varepsilon\mathbf{p})\right) / 2\varepsilon$$

with $\varepsilon = (\varepsilon_{M})^{1/3} / \|\mathbf{p}\|$, ε_{M} = machine precision. The approximation yields higher accuracy for the evaluation of the matrix-vector product than a simpler (and cheaper) forward difference stencil but is twice as expensive. The objective of the line search is to find s such

(Notes / (cont'd)

that $f(s) = f(\mathbf{u} + s\mathbf{p})$ has decreased sufficiently. It is always possible to find such a s when **p** is a descent direction, i.e., $\nabla f^T(\mathbf{u})\mathbf{p} = f'(0) < 0$. When Newton iterations are performed with the consistent Jacobian, **p** is always a descent direction since

$$f'(0) = \nabla f^T(\mathbf{u})\mathbf{p} = \mathbf{r}^T(\mathbf{u})\mathbf{J}(\mathbf{u})\mathbf{p} = \mathbf{r}^T(\mathbf{u})\mathbf{J}(\mathbf{u})(-\mathbf{J}^{-1}\mathbf{r}) = -\mathbf{r}^T\mathbf{r} < 0$$

However, use of Quasi-Newton updates or Modified Newton Raphson iterations do not always yield a descent direction. If **p** is found not to be a descent direction, the solution is stopped and a restart is created to allow repeat of the load step with a smaller time increment δt.

Line Searches with Backtracking: The objective of the line search in this case is to find $s \in]0, 1]$ such that the Goldstein-Armijo condition

$$f(s) \le f(0) + \alpha s f'(0)$$

is satisfied. Here $\alpha \in]0, 1/2]$ and we use $\alpha = 10^{-4}$. Our implementation of the Backtracking algorithm follows essentially the one presented in [13] (see also [10]).

Line Searches with Line Minimization: The objective of the line search in this case is to find s such that $\mathbf{u} = \mathbf{u} + s\mathbf{p}$ minimizes $f(s)$ in the direction **p**. The minimization is performed with Brent's method [1]. Our implementation of Brent's method follows essentially the one presented in [13]. Line minimization searches are initiated if the relaxed convergence condition:

$$|f(s)| < \varepsilon \, |f(s=0)|$$

is satisfied. The convergence tolerance □ may be of the order of 0.5. In our implementation the value $\varepsilon = 0.5$ has been adopted.

(6) Only applicable to linear iterations.

(7) If convergence_norm = other, then convergence is checked according to

$$\left| \underset{\sim}{r}^{(i)} \right| < \left| \underset{\sim}{r}^{(0)} \right| Rtol + Atol$$

References / Bibliography

1. Brent, R.P., *Algorithms for Minimization without Derivatives*, Chapter 5, Prentice Hall, Englewood Cliffs, NJ, (1973).

2. Brodlie, K.W., Gourlay, A.R., and Greenstadt, J., "Rank-one and Rank-two Corrections to Positive Definite Matrices Expressed in Product Form," *J. Inst. Maths Applics*, **11**, (1973), 73-82.

3. Broyden, C.G. *Mathematics of Computation*, **19**, (1965), 577-593.

4. Dahlquist, G. and A. Bjork, *Numerical Methods*, Prentice Hall, Englewood Cliffs, NJ, (1974).

5. Dennis, J.E. and J.J. More, "Quasi-Newton Methods, Motivation and Theory," *SIAM Rev.*, **19**, (1977), 46-89.

6. Dennis, J.E. and R.B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice Hall, Englewood Cliffs, NJ, (1983).

7 Fletcher, R., *Practical Methods of Optimization*, John Wiley, (2nd edition), (1987).

8. Gill, P.E. and W. Murray, *Numerical Methods for Constrained Optimization*, Academic Press, London, (1974).

9. Gill, P.E., W. Murray and M.H. Wright, *Practical Optimization*, Academic Press, London, (1981).

10. Johan, Z., T.J.R. Hughes and F. Shakib, "A Globally Convergent Matrix-Free Algorithm for Implicit Time-Marching Schemes Arising in Finite Element Analysis in Fluids," *Comp. Meth. Appl. Mech. Eng.*, **87**, (1991), 281-304.

11. Luenberger, D.G., *Linear and Nonlinear Programming*, Addison-Wesley, (2nd edition), (1984).

12. Matthies, H. and G. Strang, "The Solution of Nonlinear Finite Element Equations," *Int. J. Num. Meth. Eng.*, **14**, (1979), 1613-1626.

13. Press, W.H., S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, *Numerical Recipes: The Art of Scientific Computing*, Cambridge University Press, (2nd edition), (1992).

14. Sherman, J. and W. J. Morrison, "Adjustment of an Inverse Matrix Corresponding to Changes in the Elements of a Given Column or a Given Row of the Original Matrix," *Ann. Math. Stat.*, **20**, (1949), 621.

15. Woodbury, M., *Inverting Modified Matrices*, Memorandum 42, Statistics Research Group, Princeton, NJ, (1950).

Notes . .

12.4 Linear Equation Solver Selection

LINEAR_SOLVER

LINEAR_SOLVER solver_name = list , etc...

Specify the type of linear equation solver to use for all equations belonging to any given solution stagger.

Note	Variable Name	Type	Default	Description
(1)	Solver_type **direct** / iterative	list	[direct]	Solver type
(2)	Symmetric_matrix **on** / off	list	[on]	Symmetric matrix option
	Non_symmetric_matrix on / **off**	list	[off]	Nonsymmetric matrix option
	Solver_name	list	[*]	Linear equation solution procedure (see Table 1)

• *Direct Solvers*

SYMMETRIC SOLVERS

Crout_column	Crout column solver
Crout_block	Crout block solver
Frontal	Frontal solver
Cholesky_1_sparse	Sparse Cholesky with minimum degree ordering.
Cholesky_2_sparse	Sparse Cholesky with general quotient minimum degree ordering.
Cholesky_3_sparse	Sparse Cholesky with nested direction ordering.

NONSYMMETRIC SOLVERS

Crout_column	Crout column solver

(cont'd)

(cont'd)

Note	Variable Name	Type	Default	Description
	• *Iterative Solvers*			
				SYMMETRIC SOLVERS
	CG_EBE			Conjugate gradient
	CG_Crout_EBE			Conjugate gradient (Crout preconditioner)
	CG_Crout_GS_EBE			Conjugate gradient (Crout Gauss-Seidel preconditioner)
	CG_LU_EBE			Conjugate gradient (LU preconditioner)
	CG_LU_GS_EBE			Conjugate gradient (LU Gauss-Seidel preconditioner)
	CG_sparse			Conjugate gradient
	CG_IC_sparse			Conjugate gradient (incomplete Cholesky)
	CG_Mf			Conjugate gradient
	CG_Shanno_Mf			Shanno conjugate gradient
	CG_Beale_Shanno_Mf			Shanno CG with Beale's restart
				NONSYMMETRIC SOLVERS
	GMRES_EBE			GMRES
	GMRES_LU_EBE			GMRES (LU preconditioner)
	GMRES_LU_GS_EBE			GMRES (LU Gauss-Seidel preconditioner)
	GMRES_sparse			GMRES
	GMRES_ILU_sparse			GMRES (incomplete LU)
	GMRES_Mf			GMRES
	BCG_EBE			Biconjugate gradient
	BCG_sparse			Biconjugate gradient
	BCG_ILU_sparse			Biconjugate gradient (incomplete LU)
	BCG_Mf			Biconjugate gradient
	BCGS_sparse			Squared biconjugate gradient
	BCGS_ILU_sparse			Squared biconjugate gradient (incomplete LU)
	Jacobi_sparse			Jacobi iterations
	Gauss_Seidel_sparse			Gauss-Seidel iterations
	ILU_sparse			Incomplete LU iterations
	Broyden_Mf			Broyden iterations
	• *MPI Solvers*			
	MPI_Cholesky			Direct Cholesky solver
	MPI_FGMRES			Iterative GMRES solver

(cont'd)

(cont'd)

Note	Variable Name	Type	Default	Description
• *Direct Profile Solver Options*				
	Profile_minimizer	list	[on]	Equation numbering profile minimizer
	on / off			
• *Iterative Solver Options*				
	Preconditioner_type	list	[*]	Type of global preconditioner to use
	diagonal			Nodal diagonal scaling
	block_diagonal			Nodal block-diagonal scaling
• *Conjugate Gradient Solver Options*				
	Max_cg_iterations	integer	[Neq]	Maximum number of iterations
	ILU_level	integer	[25]	Incomplete LU factorization level
	Convergence_tol_cg	real	[1.E-8]	Convergence tolerance
• *GMRES Solver Options*				
	Max_gmres_iterations	integer	[Neq]	Maximum number of iterations
	Max_gmres_cycles	integer	[10]	Maximum number of Krylov subspace restart
	Num_gmres_iterations	integer	[20]	Dimension of Krylov subspace
	Convergence_tol_gmres	real	[1.E-8]	Convergence tolerance
• *MPI Iterative Solver Options*				
	Preconditioner_type	list	[*]	Type of global preconditioner
	add_schwarz			
	left_schur			
	right_schur			
	multic_schwarz			

EXAMPLE

```
Linear_Solver  /
   Solver_name = CG_Crout_EBE  /   # Conjugate gradient solver with Crout
                                   # Element by element preconditioner request
   Symmetric_matrix = on  /        # Symmetric matrix
   Preconditionner_type = diagonal  /  # Diagonal preconditionning
   Convergence_tol = 1.e-6         # Specified convergence tolerance for CG
                                         iterations
```

Table 12.4. Linear Solvers

Isolv	Solver_name	Type	Isymm*	Lhs_storage	Description
0	Crout_column	direct	0, 1	profile	Crout column solver
1	Crout_block	direct	0	out_of_core	Crout block solver
2	Frontal	direct	0	out_of_core	Frontal solver
29	Cholesky_1_sparse	direct	0	sparse	Sparse Cholesky with multiple minimum degree ordering.
30	Cholesky_2_sparse	direct	0	sparse	Sparse Cholesky with general quotient minimum degree ordering.
31	Cholesky_3_sparse	direct	0	sparse	Sparse Cholesky with nested direction ordering.
3	CG_EBE	iterative	0	element_by_element	Conjugate gradient
4	CG_Crout_EBE	iterative	0	element_by_element	Conjugate gradient (Crout preconditioner)
5	CG_Crout_GS_EBE	iterative	0	element_by_element	Conjugate gradient (Crout Gauss-Seidel preconditioner)
6	CG_LU_EBE	iterative	0	element_by_element	Conjugate gradient (LU preconditioner)
7	CG_LU_GS_EBE	iterative	0	element_by_element	Conjugate gradient (LU Gauss-Seidel preconditioner)
8	GMRES_EBE	iterative	0, 1	element_by_element	GMRES
9	GMRES_LU_EBE	iterative	0, 1	element_by_element	GMRES (LU preconditioner)
10	GMRES_LU_GS_EBE	iterative	0, 1	element_by_element	GMRES (LU Gauss-Seidel preconditioner)
11	BCG_EBE	iterative	0, 1	element_by_element	Biconjugate gradient

(cont'd)

Table 12.4. Linear Solvers (cont'd)

Isolv	Solver_name	Type	Isymm*	Lhs_storage	Description
12	CG_Mf	iterative	0	matrix_free	Conjugate gradient
13	CG_Shanno_Mf	iterative	0	matrix_free	Shanno conjugate gradient
14	CG_Beale_Shanno_Mf	iterative	0	matrix_free	Shanno CG with Beale's restart
15	BCG_Mf	iterative	0, 1	matrix_free	Biconjugate gradient
16	GMRES_Mf	iterative	0, 1	matrix_free	GMRES
17	Broyden_Mf	iterative	0, 1	matrix_free	Broyden iterations
18	CG_sparse	iterative	0	sparse	Conjugate gradient
19	CG_IC_sparse	iterative	0	sparse	Conjugate gradient (incomplete Cholesky)
20	BCG_sparse	iterative	0, 1	sparse	Biconjugate gradient
21	BCG_ILU_sparse	iterative	0, 1	sparse	Biconjugate gradient (incomplete LU)
22	BCGS_sparse	iterative	0, 1	sparse	Squared biconjugate gradient
23	BCGS_ILU_sparse	iterative	0, 1	sparse	Squared biconjugate gradient (incomplete LU)
24	GMRES_sparse	iterative	0, 1	sparse	GMRES
25	GMRES_ILU_sparse	iterative	0, 1	sparse	GMRES (incomplete LU)
26	Jacobi_sparse	iterative	0, 1	sparse	Jacobi iterations
27	Gauss_Seidel_sparse	iterative	0, 1	sparse	Gauss-Seidel iterations
28	ILU_sparse	iterative	0, 1	sparse	Incomplete LU iterations

Notes /
* Isymm = 0 Solver applicable to symmetric matrices
Isymm = 1 Solver applicable to nonsymmetric matrices

Notes / (cont'd)

(1) The application of the finite element discretization to the governing equation(s) of a field theory generates a matrix system of equations. For instance, nonlinear transient finite element dynamics are characterized by the following second-order semi-discrete balance equation:

$$\mathbf{Ma} + \mathbf{N(v, d)} = \mathbf{f} \tag{1}$$

where **M** is the mass matrix, **N** is the vector of internal forces and **f** = **f**(t) is the time-dependent external force vector. In Eq. 1, **d**, **v**, and **a** are the vectors of nodal displacements, velocities and accelerations, respectively. Eq. 1 is solved by applying a step-by-step time integration procedure resulting in a system of nonlinear algebraic equations. The solution of this system is obtained by using Newton-like iterations or related schemes. At the heart of this iteration is a set of linear equations

$$\mathbf{Ax} = \mathbf{b} \tag{2}$$

where **x** is the correction to the approximate solution vector in the nonlinear iteration loop. The matrix **A** is definite and sparse (but not necessarily always symmetric), and for the case of Eq. 1 is obtained as

$$\mathbf{A} = \mathbf{M} + \alpha\Delta t\mathbf{C} + \beta\Delta t^2\mathbf{K} \tag{3}$$

where α and β are algorithmic parameters [see Section 12.3], and [see Section 12.4]

$$\mathbf{C} = \frac{\partial \mathbf{N}}{\partial \mathbf{v}} \qquad \mathbf{K} = \frac{\partial \mathbf{N}}{\partial \mathbf{d}} \tag{4}$$

DYNAFLOW provides the capability of solving this system (Eq. 2) by either direct or iterative methods. Direct methods include:

- Profile in-core Crout
- Profile out-of-core Crout block
- Frontal in-core / out-of-core
- Explicit

and iterative methods include:

- Conjugate Gradient (CG)
- Generalized Minimum Residual (GMRES)

Notes (cont'd)

Direct solvers have the advantage of being relatively simple and robust; solutions are obtained in a predictable, finite number of operations. However, in large three-dimensional Notes / (cont'd)

simulations, storage requirements and the number of floating-point operations can become prohibitive.

These problems can be alleviated by using iterative solvers. However, iterative solvers are not necessarily robust, and preconditioning matrices are often needed to counteract problems of ill-conditioning.

Direct Methods:

Direct methods are well established and documented in standard tests (see e.g. [1, 4, 6]). In order to generate matrices with small bandwidth and profile, a profile minimizer [5, 9] option is available.

Iterative Methods:

The convergence of both the Conjugate Gradient and GMRES algorithms is greatly influenced by the matrix condition number κ (**A**). In order to reduce this sensitivity, it is useful to *precondition* the original system by using a symmetric, positive-definite matrix. In DYNAFLOW, there are two levels of preconditioning available: (1) a global preconditioner and (2) a local preconditioner (for example, an element-by element preconditioner.) These are discussed hereafter.

1. Global Preconditioner

Two global preconditioners are available as follows:

The first preconditioner available is the nodal diagonal scaling matrix $\mathbf{W}_1$ defined as:

$$\mathbf{W}_1 \equiv \text{diag } \mathbf{A} \tag{5}$$

and we rewrite Eq. 2 in a scaled form:

$$\widetilde{\mathbf{A}}\,\widetilde{\mathbf{x}} = \widetilde{\mathbf{b}} \tag{6}$$

Notes (cont'd)

where

$$\widetilde{\mathbf{A}} = \mathbf{W}_1^{-1/2}\ \mathbf{A}\ \mathbf{W}_1^{-1/2} \tag{7}$$

$$\widetilde{\mathbf{x}} = \mathbf{W}_1^{1/2}\ \mathbf{x} \tag{8}$$

$$\widetilde{\mathbf{b}} = \mathbf{W}_1^{-1/2}\ \mathbf{b} \tag{9}$$

with this definition of $\mathbf{W}_1$, diag $\widetilde{\mathbf{A}}$ becomes an identity matrix.

The second preconditioner available is the nodal block-diagonal scaling matrix $\mathbf{W}_2$ defined as:

$$\mathbf{W}_2 \equiv \text{block\_diagonal}\ (\mathbf{A}) \tag{10}$$

where the operator "block_diagonal" assigns to $\mathbf{W}_2$ the nodal block-diagonal matrix consisting of the mxm nodal diagonal blocks of **A** where m = number of equations per node ($m \leq$ Ndof). Nodal diagonal blocks of **A** are symmetric and positive definite. Therefore, $\mathbf{W}_2$ possess a well-defined Cholesky factorization:

$$\mathbf{W}_2 = \mathbf{U}^{\mathrm{T}}\mathbf{U} \tag{11}$$

and we rewrite Eq. 2 in a scaled form:

$$\widetilde{\mathbf{A}}\ \widetilde{\mathbf{x}} = \widetilde{\mathbf{b}} \tag{12}$$

where:

$$\widetilde{\mathbf{A}} = \mathbf{U}^{-\mathrm{T}}\mathbf{A}\mathbf{U}^{-1} \tag{13}$$

$$\widetilde{\mathbf{b}} = \mathbf{U}^{-\mathrm{T}}\mathbf{b} \tag{14}$$

$$\widetilde{\mathbf{x}} = \mathbf{U}\mathbf{x} \tag{15}$$

With this definition of $\mathbf{W}_2$, block_diagonal $(\widetilde{\mathbf{A}})$ becomes an identity matrix. Note that if **A** is symmetric and positive-definite, $\widetilde{\mathbf{A}}$ inherits these properties. An important attribute of block-diagonal scaling is its ability to nondimensionalize the original system of equations. This is not necessary if the system is dimensionally homogeneous. However, when this is not the case (for example, when using structural elements with both translational and rotational degrees of freedom), the 2-norm of the residual is not well-defined. To ensure against this, **A** can be scaled

Notes (cont'd)

at the element level. That is, each element matrix $\mathbf{A}^e$ is replaced with, $\tilde{\mathbf{A}}^e = \mathbf{U}^{-T}\mathbf{A}^e\mathbf{U}^{-1}$ where **U** is the Cholesky factor of $\mathbf{W}_2$.

Remark: The nodal diagonal scaling or nodal block-diagonal scaling is embedded into the solution algorithm *before* entering either the preconditioned Conjugate Gradient or the preconditioned GMRES iterative process. This requires forming $\tilde{\mathbf{b}} = \mathbf{U}^{-T}\mathbf{b}$ before entering either process and subsequently recovering $\mathbf{x} = \mathbf{U}^{-1}\tilde{\mathbf{x}}$ after the iterative solution $\tilde{\mathbf{x}}$ has been obtained. Since nodal diagonal or nodal block-diagonal scaling is *always* used, this global preconditioniong is often called pre-preconditioning or simply scaling, to distinguish it from the (optional) preconditioning used in the Conjugate Gradient or GMRES iterations. Also, in the following to simplify the notation the tilde is dropped on **A**, **b** and **x**, and the *scaled* system is written hereafter as:

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{16}$$

2. Conjugate Gradient

The Conjugate Gradient method [7] is restricted to linear systems **Ax** = **b** for which **A** is **symmetric positive-definite**. This is typically the case in solid mechanics problems. Solving such systems can be viewed as minimizing the associated potential function

$$\phi(\mathbf{x}) = 1/2\mathbf{x}^T\mathbf{A}\mathbf{x} - \mathbf{x}^T\mathbf{b} \tag{17}$$

over all $\mathbf{x} \in \mathbf{R}^N$, where N = number of equations (**A** is an NxN matrix.) An iterative strategy that begins with an initial vector $\mathbf{x}_0$ and computes a series of approximate solutions $\{\mathbf{x}_1, \mathbf{x}_2,...\}$ which progressively reduce ϕ, is used. The essence of the Conjugate Gradient Method consists of obtaining a new vector $\mathbf{x}_{i+1}$ from $\mathbf{x}_i$ along a direction $\mathbf{p}_i$ at a distance α_i:

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha_i\mathbf{p}_i \tag{18}$$

The direction vectors $\mathbf{p}_i$ are mutually conjugate and satisfy the relationships:

$$\mathbf{p}_i\mathbf{A}\mathbf{p}_j = 0 \quad \text{for all } i \neq j \tag{19}$$

and

$$\mathbf{p}_{i+1} = \mathbf{r}_i + \beta_i\mathbf{p}_i \qquad \mathbf{r}_i = \mathbf{b} - \mathbf{A}\mathbf{x}_i \tag{20}$$

The step length α_i is fixed by the condition $\partial\phi(\mathbf{x}_i + \alpha_i\mathbf{p}_i)/\partial\alpha_i = 0$, while the parameter β_i is determined by the **A**-conjugate property of the search direction $\mathbf{p}_i$. The algorithm is summarized as follows:

Notes (cont'd)

Given:

scaled linear system:	$\mathbf{Ax} = \mathbf{b}$
preconditioner:	$\mathbf{B}$
symmetric positive-definite:	$\mathbf{A}$ and $\mathbf{B}$

Initialize:

$$\mathbf{x}_0 = 0 \qquad \mathbf{r}_0 = \mathbf{b} \qquad \mathbf{p}_0 = \mathbf{z}_0 = \mathbf{B}^{-1}\mathbf{r}_0$$

Iterate:

do $i = 0, 1, 2, ..., i_{max}$

$$\alpha_i = \frac{\mathbf{r}_i^T \mathbf{z}_i}{\mathbf{p}_i^T \mathbf{A}\mathbf{p}_i}$$

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha_i \mathbf{p}_i$$

$$\mathbf{r}_{i+1} = \mathbf{r}_i - \alpha_i \mathbf{A}\mathbf{p}_i$$

Check convergence: if $\left(\|\mathbf{r}_{i+1}\|\right) \le \delta \|\mathbf{r}_0\|$ return

$$\mathbf{z}_{i+1} = \mathbf{B}^{-1}\mathbf{r}_{i+1}$$

$$\beta_i = \frac{\mathbf{r}_{i+1}^T \mathbf{z}_{i+1}}{\mathbf{r}_i^T \mathbf{z}_i}$$

$$\mathbf{p}_{i+1} = \mathbf{z}_{i+1} + \beta_i \mathbf{p}_i$$

enddo

In the above, δ = convergence_tol is a user-specified convergence tolerance, and **B** a symmetric positive-definite conditioner matrix. Similar to scaling, this second preconditioning scheme also employs a symmetric positive-definite conditioner matrix **B**, which is selected such that (1) **B** significantly contracts the spectrum of **A**, i.e., $\kappa(\mathbf{B}^{-1}\mathbf{A}) << \kappa(\mathbf{A})$; and (2) solving the linear system **Bz** = **r** is relatively inexpensive. Of course, using **B** = **A** would yield convergence to the

Notes (cont'd)

exact solution without any iteration. But this is not considered a viable choice due to its associated computational cost. Also, using **B** = **I** = identity matrix one recovers the unpreconditioned Conjugate Gradient method. The various options for defining **B** are described later in this section.

3. GMRES

The Generalized Minimal Residual (GMRES) method of Saad and Schultz [10] can be applied to any linear system **Ax** = **b**, but is usually reserved for those systems in which **A** is *non-symmetric*. The essence of the method is to obtain a trial solution vector $\mathbf{x}_0 + \mathbf{z}$ that minimizes the 2-norm of the residual $\mathbf{r} = \mathbf{b} - \mathbf{A}(\mathbf{x}_0 + \mathbf{z})$; that is to find

$$\min_{\mathbf{z} \in K} \left\| \mathbf{b} - \mathbf{A}(\mathbf{x}_0 + \mathbf{z}) \right\| \tag{21}$$

where $\mathbf{x}_0$ is an initial guess, **z** is in the k -dimensional Krylov space K:

$$\mathbf{z} \in K \equiv \text{span}\ \{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, ..., \mathbf{A}^{k-1}\mathbf{r}_0\} \tag{22}$$

and $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$. The algorithm employs an orthonormal basis of K which is obtained by the modified Gram-Schmidt procedure, and the minimization is performed using the Q-R algorithm [9]. An excellent general reference for the numerical algebraic procedures employed is [6]. Convergence of the GMRES method is also influenced by the condition number of the coefficient matrix **A**. Condition improvements are accomplished through the use of an appropriate preconditioning matrix **B** = **LU** = a positive-definite matrix, where **L** and **U** are referred to as the left and right preconditioning matrices, respectively. The preconditioned linear system then takes the form:

$$(\mathbf{L}^{-1}\mathbf{A}\mathbf{U}^{-1})(\mathbf{U}\mathbf{x}) = (\mathbf{L}^{-1}\mathbf{b}) \tag{23}$$

It should be emphasized that the matrix $(\mathbf{L}^{-1}\mathbf{A}\mathbf{U}^{-1})$ is never formed nor stored as a single matrix. Rather, $\mathbf{L}^{-1}$ and $\mathbf{U}^{-1}$ are viewed as representing operations carried out during the GMRES solution process. The algorithm is summarized as follows:

Notes (cont'd)

1. Given:
   scaled linear system: $\mathbf{Ax} = \mathbf{b}$
   preconditioner: $\mathbf{B} = \mathbf{LU}$

2. Initialize:
   $\mathbf{x} = 0$ $\mathbf{b} = \mathbf{L}^{-1}\mathbf{b}$ $\mathbf{r} = \mathbf{b}$ $\delta = \delta_{\text{tol}}\|\mathbf{r}\|$

3. GMRES cycles:

   do $l = 0, 1, 2, ..., l_{max}$ (GMRES cycles loop)

   $\mathbf{u}_1 = \mathbf{b} - \mathbf{L}^{-1}\mathbf{A}\mathbf{U}^{-1}\mathbf{x}$

   $\mathbf{e}_1 = \|\mathbf{u}_1\|$

   $$\mathbf{u}_1 = \frac{\mathbf{u}_1}{\|\mathbf{u}_1\|}$$

   do $i = 0, 1, 2, ..., k_{max}$ (GMRES iterations loop)

   $\mathbf{u}_{i+1} = \mathbf{L}^{-1}\mathbf{A}\mathbf{U}^{-1}\mathbf{u}_i$

   - do $j = 1,..., i$ (Modified Gram-Schmidt Orthogonalization)

     $\beta_{i+1,j} = \mathbf{u}_{j+1}^{T}\,\mathbf{u}_j$

     $\mathbf{u}_{i+1} = \mathbf{u}_{i+1} - \beta_{i+1,j}\,\mathbf{u}_j$

   enddo on j

   - $\mathbf{h}_i = \left\{\beta_{i+1,j}, \ldots, \beta_{i+1,i}, \left\|\mathbf{u}_{j+1}^{T}\mathbf{u}_j\right\|\right\}^{T}$
   - $$\mathbf{u}_{i+1} = \frac{\mathbf{u}_{i+1}}{\|\mathbf{u}_{i+1}\|}$$
   - Update Q-R factorization of $\mathbf{H}_i = [\mathbf{h}_1, \ldots, \mathbf{h}_i]$
   - Check convergence:

     If $(|\mathbf{e}_{i+1}| \le \delta)$ exit GMRES iteration loop i

Notes (cont'd)

enddo on i (End GMRES iterations loop)

Solve: $\mathbf{H}_i\,\mathbf{y} = \mathbf{e}$ (Using Factorized $\mathbf{H}_i$)

Update solution: $\mathbf{x} = \mathbf{x} + \sum_{j=1}^{i} y_j \mathbf{u}_j$

Check convergence:

If $\left(\left|\mathbf{e}_{i+1}\right| \le \delta\right)$ exit GMRES cycles loop l

enddo on l (End GMRES cycles loop)

4. Recover scaled system solution vector: $\mathbf{x} = \mathbf{U}^{-1}\mathbf{x}$

The GMRES iteration loop consists of formation of an orthogonal basis for the Krylov space by the modified Gram-Schmidt procedure; triangularization of the Hessenberg matrix $\mathbf{H}_i$ by the Q-R algortihm, and backsubstitution. The maximum number of GMRES cycles allowed is l_{max} = max_gmres_cycles, a user-specified value. The dimension of the Krylov space k = num_gmres_iterations, is a user-specified value, as is the convergence tolerance δ_{tol} = convergence_tol.

4. Implementation Issues

Central to both CG and GMRES methods is the computation of the matrix-vector product **Ap** of the global matrix **A** by a vector **p**, at every iteration. A naive way of performing Ap-products would be to initially compute and store all the entries in the global matrix **A** and as needed, compute explicitly the product for each vector **p**. This obviously would entail extremely large costs both with regard to computation and storage requirements. However, more efficient methods for handling Ap-products that alleviate these problems are available. DYNAFLOW provides two such methods: (1) element-by-element (EBE) storage of **A**; and (2) direct matrix-free computations of the Ap-product. In the first option, **A** is stored as a series of unassembled element matrices, whereas in the second option, **A** is never formed as discussed hereafter.

Element-by-Element (EBE) Ap-product: The method is due to Fox and Stanton [2] and Fried [3]. The matrix **A** is stored in terms of unassembled element matrices $\mathbf{A}^e$, and the Ap-product is computed via the decomposition:

$$\mathbf{Ap} = \sum_{e=1}^{Nel} \mathbf{A}^e \mathbf{p}^e \qquad (24)$$

Notes (cont'd)

where Nel = number of elements. Let Nee = number of element equations (Nee = Ned x Nen, where Ned = number of element degrees of freedom; Nen = number of nodes per element), then the data structure requires:

for symmetric $\mathbf{A}^e$:	Nee x (Nee+1)/2	words
for nonsymmetric $\mathbf{A}^e$:	Nee x Nee	words

and the calculation in Eq. 24 requires at most Nel x Nee^2 floating-point operations (half for symmetric matrices.)

Matrix-free Ap-product: In the matrix-free method, evaluation and storage of the components of **A** are avoided altogether, and an explicit forward differencing scheme is used to approximate the product. This is possible because **A** is a Jacobian-like matrix [see Section 12.3], thus a residual-like vector **r** can always be defined such that:

$$\mathbf{A}(\mathbf{u})\mathbf{p} \approx \frac{\mathbf{r}(\mathbf{u}+\varepsilon\mathbf{p}) - \mathbf{r}(\mathbf{u})}{\varepsilon} \tag{25}$$

where **u** is the solution vector, and ε a "small" number related to ε_m = computing machine precision. The vector $\mathbf{r}(\mathbf{u} + \varepsilon\ \mathbf{p})$ is sometimes referred to as the perturbed residual. Criteria for determining an optimum value for ε have been implemented in DYNAFLOW (see Johan [8] for details.)

5. Preconditioning

In the preconditioned Conjugate Gradient algorithm, the preconditioning matrix **B** is directly involved in the solution of the system **Bz** = **r**. In the GMRES algorithm, the left and right preconditioning matrices **L** and **U** obtained from **B** = **LU** are involved in the computation of $\mathbf{L}^{-1}\mathbf{A}\mathbf{U}^{-1}\mathbf{p}$ terms. Two types of preconditioning are available in DYNAFLOW: (1) element-by

element (EBE) preconditioners; and (2) global incomplete factorization-based ILU preconditioners.

Element-by-Element (EBE) Preconditioning: The terminology refers to the fact that the preconditioners are constructed from the individual element matrices $\mathbf{A}^e$ (recall that $\mathbf{A}^e$ is the e^{th} element's *scaled* contribution to the *scaled* global matrix **A**). The choice of element-by-element preconditioners is motivated by the element-based local data structure inherent to finite element codes. Two kinds of EBE preconditioners are available: (1) factorized, and (2) Gauss-Seidel. Both preconditioners are *product* decomposition approximations of the global matrix **A**. Since the matrices $\mathbf{A}^e$ are usually rank-deficient, some regularization of $\mathbf{A}^e$ is needed to remove the

Notes (cont'd)

zero-eigenvalues. This is accomplished via Diagonal or Winget [12] regularization, where the regularized matrix $\tilde{\mathbf{A}}^e$ is computed as:

$$\tilde{\mathbf{A}}^e = \mathbf{I} + \left(\mathbf{A}^e - \text{diag } \mathbf{A}^e\right) \tag{26}$$

The EBE preconditioners are constructed from the regularized element matrices $\tilde{\mathbf{A}}^e$. Details can be found in [11, 12] for both symmetric and nonsymmetric matrices.

Global Incomplete Factorization-Based (ILU) Preconditioning: An alternative preconditioning method is based on an incomplete factorization of **A**. The goal is to obtain a reasonably accurate factorization of **A** without having to form the full **A** or generating too many fill-ins. In the current implementation, an incomplete factorization by "position" has been adopted, in which the user may select the maximum height (user-specified ILU_level) of the columns used to store the incomplete factors of **A**. Then for ILU_level = 0 only the diagonal of **A** is used to form **B**, whereas for ILU_level > 0 terms in the j column of **A** from row i = j (diagonal) up to i = j - ILU_level are used to form **B**.

(2) For certain problems, a nonsymmetric matrix may be required. See element data in Chapter 9 and material data in Chapter 10 for details.

References / Bibliography

1. Duff, I.S., A.M. Erisman and J.K. Reed, *Direct Methods for Sparse Matrices*, Oxford Science Publications, Clarendon Press, Oxford, England, (1989).

2. Fox, R.L. and E.L. Stanton, "Developments in Structural Analysis by Direct Energy Minimization," *AIAA Journal*, **6**, (1968), 1036-1042.

3. Fried, I., "More on Gradient Iterative Methods in Finite-Element Analysis," *AIAA Journal*, **7**, (1969), 565-567.

4. George, J.A. and J.W-H. Liu, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice Hall, (1981).

5. Gibbs, N.E., W.G. Poole Jr., and P.K. Stockmeyer, "An Algorithm for Reducing the Bandwidth and Profile of a Sparse Matrix," *SIAM J. Num. Anal.*, **13**, (1976), 236-250.

6. Golub, G.H. and C.F. VanLoan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, (2nd edition), (1989).

7. Hestenes, M.R. and E. Steifel, "Method of Conjugate Gradients for Solving Linear Systems," *J. Research National Bureau of Standards*, **49**, (1952), 409-436.

8. Johan, Z., "Data Parallel Finite Element Techniques for Large Scale Computational Fluid Dynamics," Ph.D. Thesis, Stanford University, Stanford, CA, (1992).

9. Lewis, J.G., "Implementation of the Gibbs-Poole-Stockmeyer and Gibbs-King Algorithms," *ACM Trans. Math. Softw.*, **8**, 2, (1982), 180-189.

10. Saad, Y. and M.H. Schultz, "GMRES: A generalized minimum residual algorithm for solving nonsymmetric linear systems," S*IAM J. Sci. Statist. Comp.*, **7**, (1986), 856-869.

11. Shakib, F., T.J.R. Hughes and Z. Johan, "A Multi-Element Group Preconditioned GMRES Algorithm for Nonsymmetric Systems arising in Finite Element Analysis," *Comp. Meth. Appl. Mech. Eng.*, **75**, (1989), 415-456.

12. Shanno, D.F., "On the Convergence of a New Conjugate Gradient Algorithm", *SIAM J. Numer. Anal.,* **15**, No. 6, (1978), 1247–1257.

13. Shanno, D.F., "Conjugate Gradient Methods with Inexact Searches", *Mathematics of Operations Research,* **3**, No. 3, (1978), 244–256.

14. Winget, J.M. and T.J.R. Hughes, "Solution Algorithms for Nonlinear Transient Heat Conduction Analysis Employing Element-by-Element Strategies," *Comp. Meth. Appl. Mech. Eng.*, **52**, (1985), 711-815.

Notes . .

Notes . .

12.5 Eigenvalue Solution

EIGENVALUE_SOLUTION

EIGENVALUE_SOLUTION Eigensolver_name = list, etc...

Specify the type of eigenvalue solver to use for all equations in all subdomains/element groups belonging to any given solution stagger.

Note	Variable Name	Type	Default	Description
	Number_of_eigenpairs	integer	[0]	Number of eigenpairs requested
	Step_number	integer	[0]	Step number at which eigensolution is to be initiated.
	Frequency	integer	[1]	Eigensolution frequency
(1)	Eigensolver_name determinant_Search subspace_Iterations **block_Lanczos_num_type** block_Lanczos_int_type	list	[*]	Eigenvalue solution procedure: Determinant Search solver Subspace Iterations solver Block Lanczos solver; number type problem Block Lanczos solver; interval type problem
	Equation_type elliptic parabolic **hyperbolic**	list	[*]	Equation type (see Chapter 9 for details) Elliptic problem Parabolic problem Hyperbolic problem
• *Block Lanczos Interval Type Problems*				
	Lowest_eigenvalue	real	[0.0]	Specify lowest eigenvalue of the interval
	Highest_eigenvalue	real	[0.0]	Specify highest eigenvalue of the interval
• *Block Lanczos Number Type Problems*				
	Select_lowest_eigen **on** / off	list	[on]	Compute lowest eigenvalues
	Select_highest_eigen on / **off**	list	[off]	Compute highest eigenvalues
	Spectrum_inversion **on** / off	list	[on]	Spectrum inversion option
• *Block Lanczos Option*				
	Diagonal_mass **on** / off	list	[on]	Diagonal mass matrix
	Consistent_mass on / **off**	list	[off]	Consistent mass matrix

Notes/
(1) Implementation restricted to symmetric eigenvalue problems.

References / Bibliography

1. Bathe, K.J. and E.L. Wilson, *Numerical Methods in Finite Element Analysis*, Prentice Hall, Englewood Cliffs, NJ, (1976).

2. Parlett, B.N., *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, N.J., (1980).

3. Parlett, B.N. and D.S. Scott,"The Lanczos Algorithm with Selective Orthogonalization," *Mathematics of Computation*, **33**, 145, (1979), 217-238.

12.6 Partitioning Requests

PARTITIONS

	PARTITIONS	number_of_partitions = nparts, etc...

Specify the number of partitions and the partitioning algorithm for multi-processing using MPI implementation.

Note	Variable Name	Type	Default	Description
	Partitioning_algorithm **regions** spectral metis_nodal metis_dual	list	[*]	Partitioning algorithm
	Number_of_partitions	integer	[0]	Number of partitions
	Overlapping **on** / off	list	[on]	Overlapping option

Notes/

(1) If greater than 1, defaults to number_of_processors during mpi execution.

Notes . .

12.7 Initialization Requests

12.7.1 Initialize Displacement Array(s)

INITIALIZE_D0

INITIALIZE_D0 number_of_load_steps = nstep, etc...

For certain problems, it is required that an initialization takes place (e.g., for pressure dependent materials which require that gravity induced initial stresses be first computed, or for prestressing structural members). In that case nstep load steps are used to establish initial equilibrium conditions. By default, the displacements are thereafter set to zero, e.g. $\mathbf{d}_0 = \mathbf{0}$.

Note	Variable Name	Type	Default	Description
	Stagger_name(s)	list	[all]	Stagger names to be initialized. Name(s) must be enclosed in quotation marks.
	Stagger_number(s)	integer list	[all]	Stagger numbers to be initialized
	Steady_state **on** / off	list	[on]	Steady state option
	Number_of_load_steps	integer	[1]	Number of load steps to be used for initialization
	Reform_lhs_freq	integer	[0]	Number of load steps between stiffness reform
(1)	Clear_array **on** / off	list	[on]	clear $\mathbf{d}_0$ array option.
	Clear_stagger_name(s)	list	[all]	Stagger names to be cleared. Name(s) must be enclosed in quotation marks.
	Clear_stagger_number(s)	integer	[all]	Stagger numbers to be cleared.
	Xfem_option **on** / off	list	[on]	Xfem option flag

Notes/
(1) Not clearing $\mathbf{d}_0$ may affect convergence during the initialization phase.

12.7.2 Initialize Velocity Array(s)

INITIALIZE_V0

INITIALIZE_V0

The default is $\mathbf{v}_0 = \mathbf{0}$. However, for certain problems, it may be required that an initialization takes place for $\mathbf{v}_0$ (Parabolic initial BVP), viz.,

$$\mathbf{M}\,\mathbf{v}_0 = \mathbf{f}_0 - \mathbf{K}\,\mathbf{d}_0$$

12.7.3 Initialize Acceleration Array(s)

INITIALIZE_A0

INITIALIZE_A0

The default is $\mathbf{a}_0 = \mathbf{0}$. However, for certain problems, it may be required that an initialization takes place for $\mathbf{a}_0$ (Hyperbolic initial BVP), viz.,

$$\mathbf{M}\,\mathbf{a}_0 = \mathbf{f}_0 - \mathbf{C}\,\mathbf{v}_0 - \mathbf{K}\,\mathbf{d}_0$$

Notes . .

Notes . .

12.8 Meshing Requests

12.8.1 Update Nodal Coordinates Requests

UPDATE_COORDINATES

UPDATE_COORDINATES	step_number = ns, etc...
Update coordinates starting at step number ns.	

Note	Variable Name	Type	Default	Description
	Step_number	integer	[0]	Step number
	Frequency	integer	[0]	Number of time steps between coordinates update
(1)	Stagger_name(s)	string	[all]	Solution staggers for which the coordinate array is to be updated. Name(s) must be enclosed in quotation marks.

Notes/
(1) List all solution staggers for which the coordinate array is to be updated.

12.8.2 Remeshing Requests

REMESH

	REMESH step_number = ns, etc...
	Remesh starting at step number ns.

Note	Variable Name	Type	Default	Description
	Algorithm **Laplacian** Tim_Baker fusing other	list	[*]	Algorithm
	Step_number	integer	[0]	Time step number at which remeshing is to be initiated
	Frequency	integer	[0]	Step number frequency at which remeshing is to be initiated
(1)	Region(s)	string	[none]	Regions for which remeshing is to be performed Name(s) must be enclosed in quotation marks
(1)	Element_group(s)	string	[none]	Element groups for which remeshing is to be performed Name(s) must be enclosed in quotation marks

EXAMPLE

```
Remesh /
   step_number = 1      /
   frequency = 1      /
   element_group(s) = "mesh"
```

Notes/
(1) List all region names and/or element group names for which remeshing is to be done.

Notes . .

Notes . .

12.9 Clear Requests

12.9.1 Clear Nodal Displacement Array(s) Requests

CLEAR_D

CLEAR_D step_number = ns, etc...

Reset displacement array(s) to zero starting at step number ns.

Note	Variable Name	Type	Default	Description
	Step_number	integer	[0]	Step number
	Frequency	integer	[0]	Number of time steps between clears
(1)	Stagger_name(s)	string	[all]	Solution staggers for which the displacement array is to be cleared. Name(s) must be enclosed in quotation marks.

Notes/
(1) List all solution staggers for which the displacement array is to be cleared.

12.9.2 Clear Nodal Velocity Array(s) Requests

CLEAR_V

CLEAR_V step_number = ns, etc...

Reset velocity array(s) to zero starting at step number ns.

Note	Variable Name	Type	Default	Description
	Step_number	integer	[0]	Step number
	Frequency	integer	[0]	Number of time steps between clears
(1)	Stagger_name(s)	string	[all]	Solution staggers for which the velocity array is to be cleared. Name(s) must be enclosed in quotation marks.

Notes/
(1) List all solution staggers for which the velocity array is to be cleared.

12.9.3 Clear Nodal Acceleration Array(s) Requests

CLEAR_A

	CLEAR_A step_number = ns, etc...
	Reset acceleration array(s) to zero starting at step number ns.

Note	Variable Name	Type	Default	Description
	Step_number	integer	[0]	Step number
	Frequency	integer	[0]	Number of time steps between clears
(1)	Stagger_name(s)	string	[all]	Solution staggers for which the acceleration array is to be cleared. Name(s) must be enclosed in quotation marks.

Notes/
(1) List all solution staggers for which the acceleration array is to be cleared.

12.9.4 Clear Stress Array(s) Request

CLEAR_STRESS

	CLEAR_STRESS step_number = ns, etc...

Note	Variable Name	Type	Default	Description
	Step_number	integer	[0]	Step number
	Frequency	integer	[0]	Number of time steps between clears
(1)	Element_group(s)	string	[all]	Element groups for which stress is to be cleared. Names must be enclosed in quotation marks.

Notes /
(1) List all element groups for which the stress array is to be cleared.

12.9.5 Clear Time Requests

CLEAR_T

	CLEAR_T step_number = ns, etc...
	Reset time to zero.

Note	Variable Name	Type	Default	Description
	Step_number	integer	[0]	Step number
	Frequency	integer	[0]	Number of time steps between clears

Notes . .

12.10 Error Recovery

RECOVER_ERROR

RECOVER_ERROR step_number = ns, etc....

Determine the magnitude of the error at any given time step.

Note	Variable Name	Type	Default	Description
(1)	Element_group(s)	string	[none]	Element group(s) for which the error is to be estimated. Name(s) must be enclosed in quotation marks.
	Step_number	integer	[0]	Step number at which error recovery is to be initiated.
	Frequency	integer	[0]	Error recovery frequency

EXAMPLE

```
Recover_error /
        element_group(s) = "group_1", "group_2" /
        step_number = 1 /
        frequency = 1
```

Notes/

(1) List all element group names or numbers for which the error is to be estimated.

(2) The finite element provides an approximation to the exact solution of a mathematically well-posed problem. The difference between the exact and approximate solutions decreases as the size of the subdivision h gets smaller, or as the order of the interpolation polynomial functions used increases. In order to determine exactly the magnitude of the error at a given stage of subdivision the exact solution must be available, a situation not common in practice.

Error estimates can be obtained by projecting the field quantities onto the nodes to obtain a continuous field and by comparing the projected quantities with the elemental values, e.g., the error in stresses in continuum type problems defined as

$$\mathbf{e}_\sigma = \sigma - \hat{\sigma} \qquad (1)$$

where σ = "exact solution" and $\hat{\sigma}$ = "finite element" stress, can be estimated with good accuracy as:

$$\mathbf{e}_\sigma = \sigma^* - \hat{\sigma} \qquad (2)$$

where σ^* = "projected" finite element stress. The projection is performed by using the same interpolation functions as used for representing the displacement field $\hat{\mathbf{u}}$, viz.,

$$\hat{\mathbf{u}} = \mathbf{N}\,\bar{\mathbf{u}} \tag{3}$$

where $\bar{\mathbf{u}}$ = vector of nodal displacements, and **N** = nodal interpolation function. Then

$$\sigma^* = \mathbf{N}\bar{\sigma}^* \tag{4}$$

The approximating equation is achieved by a weighted residual requirement for equality between and σ^* and $\hat{\sigma}$, viz.,

$$\int_{\Omega} \mathbf{N}^T \left(\sigma^* - \hat{\sigma}\right) \mathrm{d}\,\Omega = 0 \tag{5}$$

giving rise to the following matrix problem:

$$\bar{\sigma}^* = \mathbf{A}^{-1}\left(\int_{\Omega} \mathbf{N}^T \hat{\sigma}\ \mathrm{d}\,\Omega\right) \tag{6}$$

where

$$\mathbf{A} = \int_{\Omega} \mathbf{N}^T\ \mathbf{N}\ \mathrm{d}\,\Omega \tag{7}$$

The above computation is made particularly simple if a lumped or diagonal approximation is made to **A** by using nodal quadrature. With both σ^* and $\hat{\sigma}$ fields available, the error in stress can be estimated with good accuracy as:

$$\mathbf{e}_{\sigma} = \sigma^* - \hat{\sigma} \tag{8}$$

The error computed in this manner is an *a posterior* error estimate, since it can only be computed after the solution of the problem has been obtained. The following error norms are computed:

$$\left\|\mathbf{e}_{\sigma}\right\|_{L^2} = \left[\int_{\Omega} \left(\sigma^* - \hat{\sigma}\right)^T\ \left(\sigma^* - \hat{\sigma}\right) \mathrm{d}\,\Omega\right]^{1/2} = \mathrm{L}_2 \text{ norm of stress error} \tag{9}$$

$$\left|\Delta \mathbf{e}_{\sigma}\right| = \left(\left\|\mathbf{e}_{\sigma}\right\|^2_{\mathrm{L}2}\ /\,\Omega\right)^{1/2} = \text{RMS error in stress} \tag{10}$$

These error norms are evaluated over the whole domain and over each finite element subdomain. We note that:

$$\|\mathbf{e}\|^2 = \sum_{i=1}^{\text{Numel}} \|\mathbf{e}\|_i^2 \tag{11}$$

where i refers to the individual finite element subdomains $\Omega_i \left(\Omega = \mathop{\mathbf{U}}_{i=1}^{\text{Numel}} \Omega_i \right)$. A posterior error estimates can be used to refine the finite element approximation locally to achieve results of a given desired accuracy economically and optimally. For that purpose, the error is projected onto the nodes and may be displayed graphically.

Notes . .

12.11 Strain Energy Recovery Requests

RECOVER _STRAIN_ENERGY

	RECOVER_STRAIN_ENERGY			step_number = ns, etc...
	Compute strain energy starting at step number ns.			
Note	Variable Name	Type	Default	Description
(1)	Element_group(s)	string	[none]	Element groups for which strain energy is to be computed Name(s) must be enclosed in quotation marks
	Step_number	integer	[0]	Time step number at which computing strain energy is to beinitiated
	Frequency	integer	[0]	Step number frequency at which computing strain energy is to be performed

EXAMPLE

```
Recover_strain_energy /
   step_number  =  1     /
   frequency = 1     /
   element_group(s) = "solid"
```

Notes /

(1) List all element group names for which the strain_energy is to be computed.

Notes . .

12.12 System Compliance Requests

SYSTEM_COMPLIANCE

	SYSTEM_COMPLIANCE	step_number = ns, etc...

Compute system compliance starting at step number ns. The system compliance is defined as:

$$W = \frac{1}{2}\mathbf{d}^T\,\mathbf{K}\,\mathbf{d} = \frac{1}{2}\sum_e \mathbf{d}^{e^T}\,\mathbf{k}^e\,\mathbf{d}^e$$

where the global displacement vector **d** is the solution to the equilibrium problem:

$$\mathbf{K}\,\mathbf{d} = \mathbf{f}$$

and **K** = global stiffness matrix of the discretized system.

Note	Variable Name	Type	Default	Description
(1)	Element_group(s)	string	[none]	Element groups for which computing compliance is to be performed Name(s) must be enclosed in quotation marks
	Step_number	integer	[0]	Time step number at which computing compliance is to be initiated
	Frequency	integer	[0]	Step number frequency at which computing compliance is to be performed

EXAMPLE

```
System_compliance /
   step_number  =  1     /
   frequency = 1     /
   element_group(s) = "solid"
```

Notes /
(1) List all element group names for which the system_compliance is to be recovered.

Notes . .

12.13 Layout Optimization

LAYOUT_OPTIMIZATION

	LAYOUT_OPTIMIZATION	step_number = ns , etc...

The procedure computes the distribution of material for structure supported on its boundaries and subjected to a given loading condition, such that its compliance (objective function) is optimized. The amount of material is constrained and the spatial distribution of material is limited to the design domain. The design domain can have holes (void regions) and regions with fixed solid material. The structure can be discretized in a number of different ways (e.g., using truss, beam, plate or solid elements), and can be modeled as two- or three-dimensional.

Note	Variable Name	Type	Default	Description
	Stagger_name(s)	string	[main]	Stagger name(s). Name(s) must be enclosed in quotation marks.
	Element_group(s)	string	[all]	Element group(s) for which optimization is to be performed. Name(s) must be enclosed in quotation marks.
	Step_number	integer	[1]	Step number at which optimization is to be initiated.
	Frequency	integer	[0]	Optimization frequency
(1)	Average_density	real	[0.0]	Target average material density $\rho_{\text{average}} > 0$
	Minimum_density	real	[0.01]	Minimum density $\rho_{\min} > 0$
	Maximum_density	real	[1.0]	Maximum density $\rho_{\max} > 0$
	Initial_density **uniform** random deviates	list	[*]	Initial density uniform random deviates
	Seed	integer	[7654321]	Seed for random number generation; > 100,000 and < 1,000,000
	Extreme_size	real	[0.5]	Extreme size ≤ 0.5
	Extreme_prob	real	[0.5]	Extreme probability ≤ 1.0
(2)	Exponent	real	[1.0]	Density exponent p ≤ 3
	Damping	real	[0.5]	Damping $\eta > 0$

cont'd

(cont'd)

Note	Variable Name	Type	Default	Description
	Move_limit	real	[0.2]	Move limit
	Filter_size	real	[0.0]	Filter size
	Max_number_of_iterations	integer	[100]	Maximum number of iterations

EXAMPLE

```
Layout_Optimization  /
    stagger_name = "main"  /
    element_group(s) = "group 1 "  /
    step_number = 1  /
    average_density = 0.75  /
    minimum_density = 0.001  /
    maximum_density = 1.0  /
    exponent = 1.0  /
    damping = 0.8  /
    max_number_of_iterations = 15
```

Notes /

(1) The optimization algorithm minimizes the compliance of the structure for a fixed amount of available material in the design domain. Let ρ^e denote the density of material in element e with volume ϑ^e, then the total volume of the structure is computed as

$$V = \sum_e \rho^e \vartheta^e = \rho_{average} \sum_e \vartheta^e$$

where ρ^e is bounded by:

$$0 < \rho^e_{min} \le \rho^e \le \rho^e_{max} \le 1$$

(2) The stiffness of any given element is computed as:

$$\left[\mathbf{S}^e\right] = \left(\rho^e\right)^p \left[\mathbf{S}^e_0\right] \qquad 1 \le p \le 3$$

where $\left[\mathbf{S}^e_0\right]$ = element stiffness for solid material, and p = density exponent.

References / Bibliography

1. Bendsoe, M.P., *Optimization of Structural Topology, Shape, and Material,* Springer, (1995).

Notes . .

Notes . .

12.14 Constitutive Experiment Requests

CONSTITUTIVE_EXPERIMENT

CONSTITUTIVE_ EXPERIMENT Number_of_experiments = nexp, etc...
< experiment data >
< mode-shapes data >
< material data ⇒ only required if element_number = 0 >
< terminate with a blank record >

Perform constitutive experiments on material model.

Note	Variable Name	Type	Default	Description
	Number_of_experiments	integer	[0]	Number of constitutive experiments
(1)	Number_of_mode_shapes	integer	[0]	Number of mode shapes
	Mode_shape **strain** / stress	list	[strain]	Mode shape option: Prescribed strain or stress amplitudes
(2)	Fluid_bulk_load_time	integer	[0]	Fluid bulk load-time function number
	Print **on** / off	list	[on]	Print stress/strain amplitudes option
(3)	Element_number	integer	[0]	Element number ≥ 0
	Group_number	integer	[0]	Group Number ≥ 0 and ≤ NUMEG
	Stress_point_number	integer	[0]	Stress point number ≥ 1 and ≤ Nspts

---

EXAMPLE

```
Constitutive_Experiment /
  Number_of_experiments = 1 /          #
  Number_of_mode_shapes = 1 /          #
  Mode_shape = stress                  # prescribe stress amplitudes

Experiment  1 , 1000 , 1.00            # 1000 time steps with time step = 1.0
Mode-Shape  1 , 2 , 0. 0 , -2.e4 , 0.0 #  σ22 = -2.e4 with load-time function 2
```

Notes/

(1) The prescribed stress or strain amplitudes are prescribed by using NSHAP mode-shapes and corresponding load-time functions (Section 8.0).

(2) Corresponds to the load-time function input in Section 8.0. Only applicable to stress-driven experiments. This option allows variable draining conditions to be used (e.g., BULKF = 0.0 for drained conditions, BULKF = ALF for undrained conditions).

(3) This allows constitutive experiments to be performed on material data previously defined within existing element groups.

12.14.1 Experiments (Nexp)

EXPERIMENT

EXPERIMENT n, nts(n), dt(n)

Note	Variable	Type	Default	Description
	N	integer	[0]	Experiment number ≥ 1 and ≤ Nexp
	NTS(N)	integer	[0]	Number of time steps
	DT(N)	real	[0.0]	Time-step size

12.14.2 Mode-Shapes (Nshape)

MODE_SHAPE

MODE_SHAPE lshap, ltime, <f (i, lshap), i = 1, 6>

Note	Variable	Type	Default	Description
	LSHAP	integer	[0]	Mode shape number ≥ 1 and ≤ Nshape
	LTIME	integer	[0]	Load time function number ≥1 and ≤ NLF
(1)	F(I, LSHAP)	real	[0.0]	Mode-shape; i = 1, 6

Notes/
(1) The prescribed stress or strain amplitudes are prescribed by using load-time functions (Chapter 8) and NSHAP mode-shapes. The corresponding mode-shapes are defined for each i (i = 1, 6) as follows:

$\varepsilon_{11}, \varepsilon_{22}, \varepsilon_{33}, \gamma_{12}, \gamma_{23}, \gamma_{31}$	Strain driven case
$\sigma_{11}, \sigma_{22}, \sigma_{33}, \tau_{12}, \tau_{23}, \tau_{31}$	Stress driven case

12.14.3 Material Data (only if required)

Only required if Element_number = 0. The data for material properties must follow. Consult Chapter 10 for the required input of the individual material models.

Notes . .

Notes . .

Notes . .

Notes . .

13.0 SAMPLE DATA

```
#=============================================================================
# Example: Consolidation of a poroelastic sphere subject to impulse
#         pressure load on its outer surface boundary
#-----------------------------------------------------------------------------------------------------------
--
# Problem Description:
#         Coupled solution using QDC_solid and QDC_pressure
#-----------------------------------------------------------------------------------------------------------
--
#       number_of_nodal_points = 73
#       number_of_spatial_dimensions = 2
#       number_of_degrees_of_freedom = 3
#       number_of_element_groups = 2
#                 number_of_elements = 60
#       number_of_surface_load_cases = 1
#       number_of_time_steps = 20
#=============================================================================
PRINT
DEFINE_PROBLEM \
       name = "sphere_stagger" \
       title = " ----- Consolidation of a poroelastic sphere ------ "
#-----------------------------------------------------------------------------------------------------------
--
NODAL_COORDINATES \
  generation_type = cylindrical \
  reference_coordinates x0=0 y0=0 z0=0

         1  0 0.00  0.00
         2  4 0.50  0.00  0.00
         3  0 4.00  00.0
         4  0 4.00  90.0
         5  0 0.50  90.0
         7  1  8  8

SURFACE_LOADS \
  number_of_loads = 1 \
  geometry_type = axisymmetric

           1    1.00     1.00

     Nodal_connectivity
           1  1  9  17  1
           8  1  8
```

```
NODAL_BOUNDARY_CONDITIONS \
  variable = solid_displacement \
  generation_type = type_1

          1    0    1    1
          2    1    0    1
          9    0    0    1
         66    1    1    0
         73    0    1    0

NODAL_BOUNDARY_CONDITIONS \
  variable = pressure \
  generation_type = type_1

          9    8    1
         73    0    1

DEFINE_MATERIAL_MODEL \
      name = "dummy" \
      number_of_material_sets = 1

    Stress_Model \
      Material_type = Linear \
      Material_name = Linear_elastic

        Material_Set_Number = 1 \
        Youngs_modulus = 1.0e4 \
        Poissons_ratio = 0.25 \
        solid_mass_density = 2.0 \
        fluid_mass_density = 1.0 \
        Porosity = 0.30

    Scalar_Diffusion_Model \
      Material_type = Linear \
      Material_name = Scalar_diffusion

        Material_Set_Number = 1 \
        Mass_density = 1.0 \
        Compressibility = 0.0e-6 \
        Permeability \
          type = isotropic \
              k_11 = 1.60e-3
```

```
DEFINE_REGION \
  name = "Porous_solid" \
  element_type = continuum \
  element_shape = four_node_quad \
  analysis_type = axisymmetric \
  strain_displacement = bbar_mean \
  number_of_output_sets = 1 \
  material_model = "dummy"

    Nodal_connectivity
    1    1    2    3   11   10    1
    7    1    1    8    7    8
   57    1    1    2   10   18
   58    1    1   18   26   34
   59    1    1   34   42   50
   60    1    1   50   58   66

    Field_Output
      58    0   17

    Body_Force  b_x1=0.00  b_x2=0.00   h=0.00

  ELEMENT_GROUP \
    name = "Solid_equation" \
    element_name = QDC_solid

  ELEMENT_GROUP \
    name = "Diffusion_equation" \
    element_name = QDC_pressure

END

TRANSLATOR \
       output_format = "femgv"

MESH_DUMP /
  nodal_step =  0 /
  nodal_frequency = 5 /
  spatial_step =  0 /
  spatial_frequency = 5 /
  final_time = on
```

```
NODAL_HISTORY
   1    0    3
  35    0    3
  73    0    2

TIME_SEQUENCE \
   number_of_time_steps = 20 \
   time_step = 0.025e+0

INITIALIZE_V0

DEFINE_STAGGER name="Group1" \
   include_element_group(s) = "Solid_equation" \
   equation_type= parabolic \
   alpha=1.0 beta=0.0 \
   nonlinear_iterations \
     jacobian_matrix = on \
      max_number_of_iterations = 0 \
   variable= Solid_displacement

DEFINE_STAGGER name="Group2" \
   include_element_group(s) = "Diffusion_equation" \
   equation_type= parabolic \
   alpha=1.0 beta=0.0 \
   nonlinear_iterations \
     jacobian_matrix = on \
      max_number_of_iterations = 0 \
   variable= Pressure

STAGGER_CONTROL \
   max_number_of_iterations = 3 \
   convergence_check = on \
   convergence_tol_sol = 1.0e-4 \
   convergence_tol_cg = 1.0e-9

c PRINT_map

RUN_SOLVER \
       mode = execution
STOP
#-------------------------------------------------------------------------------------------------------------
-
```

Notes . .

Notes . .

Appendix A

TYPICAL PHYSICAL PROPERTIES OF SOME MATERIALS

Material	Density kg/m^3	Ultimate Strength Tension MPa	Ultimate Strength Comp. MPa	Yield Strength MPa	Modulus of Elasticity E GPa	Poisson's ratio	Coef. of Thermal Exp. 10^{-6}/°C	Thermal Conductivity W/m °C
Aluminum 2014-T6	2800	470		410	72	0.33	23	210
(alloy) 6061-T6	2800	228		131	70	0.33	23	210
Brass cold rolled	8470	540		420	105	0.35	19	105
annealed	8470	330		100	105	0.35	19	105
Bronze Manganese	8800	450		170	100	0.34	20	58
Cast Iron								
Gray	7200	170	650		95	0.25	12	45
Malleable	7200	370		250	170	0.25	12	45
Concrete								
Low strength	2400	2	20		22	0.15	11	1
High strength	2400	3	41		32	0.15	11	1
Medium strength	2400	4	62		40	0.15	11	1
Copper hard-drawn	8900	380		330	120	0.33	17	380
Glass Silicon	2400	80	400		70	0.17	8	0.8
Magnesium 8.5% Al	1800	350		250	45	0.35	26	160
Steel								
0.2%C HR	7850	410		250	200	0.30	12	42
0.2%C HR	7850	550		350	200	0.30	12	42
0.2%C HR	7850	690		370	200	0.30	12	42
0.8%C HR quenched	7850	830		700	200	0.30	12	42
Stainless 302 CR	7920	860		600	194	0.30	17	18
Titanium 6% Al 4%V	4460	900		830	110	0.34	9	14

Properties vary widely depending on changes in composition, temperature and treatment conditions.
CR = Cold rolled HR = Hot rolled

Appendix A

IMPORTANT CONSTANTS

Constant	U.S. Unit	SI Unit
Absolute zero	-459.67 °F	-273.15 °F
Acceleration of gravity	32.174 ft/s^2	9.8066 m/s^2
Atmospheric pressure	14.694 psi	0.10132×10^6 Pa
Stefan-Boltzmann constant	0.1714×10^{-8} Btu/hr ft^2 $°R^4$ where °R = °F + 459.67	5.669×10^{-8} W/m^2 $°K^4$ where °K = °C + 273.15

APPROXIMATE PROPERTIES OF MILD STEEL AT ROOM TEMPERATURE

Quantity	U.S. Unit	SI Unit
Conductivity	28.9 Btu/ft hr °F 2.4 Btu/in hr °F	50 W/m °C
Density	15.13 slug/ft^3(lbf s^2/ft^4) 0.730×10^{-3} lbf s^2/ft^4 0.282 lbm/in^3	7800 kg/m^3
Elastic modulus	30×10^6 psi	207×10^9 Pa
Specific heat	0.11 Btu/lbm °F	460 J/kg °C
Yield stress	30×10^3 psi	207×10^6 Pa

Appendix A

SI System – Units

Temperature	Kelvin	K	
Length	meter	m	
Time	second	s	
Mass	kilogram	kg	
Force	newton	N	$kg \cdot m/s^2$
Pressure	pascal	Pa	N/m^2
Work and energy	joule	J	$N \cdot m$
Power	watt	W	J/s
Electric current	ampere	A	
Electric charge	coulomb	C	$s \cdot A$
Electric potential	volt	V	W/A
Electric resistance	ohm	Ω	V/A
Electric conductance	seimens	S	A/V
Capacitance	farad	F	C/V
Permittivity			F/m
Magnetic flux	weber	Wb	$V \cdot s$
Magnetic flux density	tesla	T	Wb/m^2
Inductance	henry	H	Wb/A
Magnetic field			A/m
Permeability			H/m
Concentration			mol/m^3
Frequency	hertz	Hz	cycle/s

Common SI Prefixes

Tera	T	10^{12}	milli	m	10^{-3}
Giga	G	10^{9}	micro	μ	10^{-6}
Mega	M	10^{6}	nano	n	10^{-9}
Kilo	k	10^{3}	pico	p	10^{-12}

Appendix A

Constants

Quantity	Symbol	Value
Electron charge	e_0	1.602 x 10^{-19} C
Electron mass	m_e	9.1091 x 10^{-31} kg
Bohr magneton	μ_B	9.273 x 10^{-24} J/K
Boltzmann constant	k	1.381 x 10^{-23} J/K
Avogadro's number	N_{av}	6.022 x 10^{-23} mol^{-1}
Gas constant	R	8.314 J/(mol K)
Faraday's constant	F	9.6485 x 104 C/mol
Permeability of free space	μ_0	4π x 10^{-7} H/m
Permittivity of free space	ε_0	8.854 x 10^{-12} F/m
Characteristic impedance of free space	z_0	377 **Ω**
Speed of light in vacuum	c_0	2.998 x 10^{-8} m/s
Atmospheric pressure	p_{atm}	101.325 kPa
Zero degrees Celsius	0°C	273.15 K
Gravitational constant	G	6.673 x 10^{-11} $N \cdot m^2/kg^2$

Notes:

- 1Å = 1 ångström = 10^{-10} m
- Avogadro's number is the number of elementary entities in one mole. The elementary entity must be specified; it may be atoms, molecules, ions, electrons or other particles. Values found for Avogadro's number range within 1% of the listed value.
- Relationships between constants: $Re_0^2 N_{av} = kF^2$, $F = e_0 N_{av}$, $R = kN_{av}$.

Appendix B

USEFUL RELATIONSHIPS AMONG ISOTROPIC ELASTIC CONSTANTS

$$\lambda = \frac{2Gv}{1-2v} = \frac{G(E-2G)}{3G-E} = B - \frac{2}{3}G = \frac{Ev}{(1+v)(1-2v)} = \frac{3Bv}{1+v} = \frac{3B(3B-E)}{9B-E}$$

$$\mu \equiv G = \frac{\lambda(1-2v)}{2v} = \frac{3}{2}(B-\lambda) = \frac{E}{2(1+v)} = \frac{3B(1-2v)}{2(1+v)} = \frac{3BE}{9B-E}$$

$$v = \frac{\lambda}{2(\lambda+G)} = \frac{\lambda}{(3B-\lambda)} = \frac{E}{2G} - 1 = \frac{3B-2G}{2(3B+G)} = \frac{3B-E}{6B}$$

$$E = \frac{G(3\lambda+2G)}{\lambda+G} = \frac{\lambda(1+v)(1-2v)}{v} = \frac{9B(B-\lambda)}{3B-\lambda} = 2G(1+v) = \frac{9BG}{3B+G} = 3B(1-2v)$$

$$B = \lambda + \frac{2}{3}G = \frac{\lambda(1+v)}{3v} = \frac{2G(1+v)}{3(1-2v)} = \frac{GE}{3(3G-E)} = \frac{E}{3(1-2v)}$$

Also Useful Are

$$\frac{G}{\lambda+G} = 1-2v, \quad \frac{\lambda}{\lambda+2G} = \frac{v}{1-v}, \quad \frac{\lambda+2G}{E} = \frac{1-v}{(1+v)(1-2v)}, \quad \frac{E}{1-v^2} = \frac{4G(\lambda+G)}{\lambda+2G}$$

Constant Names

$\lambda,\ \mu =$ Lame's constants

$G \equiv \mu =$ Shear modulus

$B =$ Bulk modulus

$E =$ Young's modulus

$v =$ Poisson's ratio

Appendix C

UNITS, CONVERSIONS AND ABBREVIATIONS

General Prefixes

10	deka (da)	10^{-1}	deci (d)
10^{2}	hecto (h)	10^{-2}	centi (c)
10^{3}	kilo (k)	10^{-3}	milli (m)
10^{6}	mega (M)	10^{-6}	micro (μ)
10^{9}	giga (G)	10^{-9}	nano (n)
10^{12}	tera (T)	10^{-12}	pico (p)
10^{15}	peta (P)	10^{-15}	femto (f)
10^{18}	exa (E)	10^{-18}	atto (a)

Length

1 meter (m) = 100 centimeters (cm) = 3.281 feet (ft) = 39.37 inches (in)
1 mile = 5280 ft = 1.609 kilometers (km)
1 micron (μ) = 10^{-6} m
1 angstrom (Å) = 10^{-10} m

Area

1 hectare (ha) = 10^{4} square meters (m^2) = 2.47 acres
1 acre = 43,560 square feet (ft^2)
1 barn (b) = 10^{-24} cm^2

Volume

1 cubic meter (m^3) = 1000 liters = 264.2 U.S. gallons = 35.31 cubic feet (ft^3)
1 liter (1) = 10^{3} cubic centimeters(cm^3 or ml) = 1.057 U.S. quarts
1 acre foot = 1.234 x 10^{3} m^3
1 cord = 128 ft^3
1 board foot = 2.36 x 10^{-3} m^3
1 cubic mile = 4.17 cubic kilometers (km^3)
1 barrel of petroleum (bbl) = 42 U.S. gallons = 0.159 m^3

Angles

360 degrees (°) = 2π radians
1 degree = 60 minutes (') of arc
1 minute of arc = 60 seconds (") of arc

Time

1 year (y or yr) = 3.1536 x 10^{7}seconds (s or sec)
= 8.76 x 10^{3} hours (h or hr)
1 day (d) = 8.64 x 10^{4} sec = 1440 minutes (min)

Appendix C

Mass

1 kilogram (kg) = 2.205 pounds (lb)

1 metric ton (tonne or MT) = 10^3 kilograms (kg)
= 1.102 short tons
= 0.9842 long tons

1 pound (lb) = 16 ounces avoirdupois (oz) = 453.6 grams (g)

Energy

1 joule (J) = 1 kg m^2/sec^2
= 10^7 ergs = 0.2390 calories (cal)
= 9.484 x 10^{-4} British thermal units (Btu)
= 1 watt-second (Ws)
= 6.242 x 10^{18} electron volts (eV)
= 1 newton-meter (Nm)

1 kilowat-hour (kWh) = 3.6 x 10^6 J
= 3414 Btu

1 quad = 10^{15} Btu = 1.05 x 10^{18} J

1 Calorie = 1 kilocalorie (Kcal) = 10^3 cal

1 therm = 10^5 Btu

1 foot pound = 1.356 J

1 kiloton of TNT (KT) = 4.2 x 10^{12} J

Power

1 watt (W) = 1 joule/second

1 horsepower (hp) = 0.746 kilowatts (kW)

Force

1 newton (N) = 1 kg m/sec^2 = 10^5 dynes (dyn)

Pressure

1 pascal = 1 N/m^2 = 1 J/m^3

1 bar = 10^5 pascal = 0.9869 atmospheres (atm)

1 atmosphere (atm) = 76 cm of mercury
= 14.7 lb/in^2
= 760 torr

Viscosity

1 poise (p) = 1 dyn-sec/cm^2 = 0.1 kg/m sec

Permeability

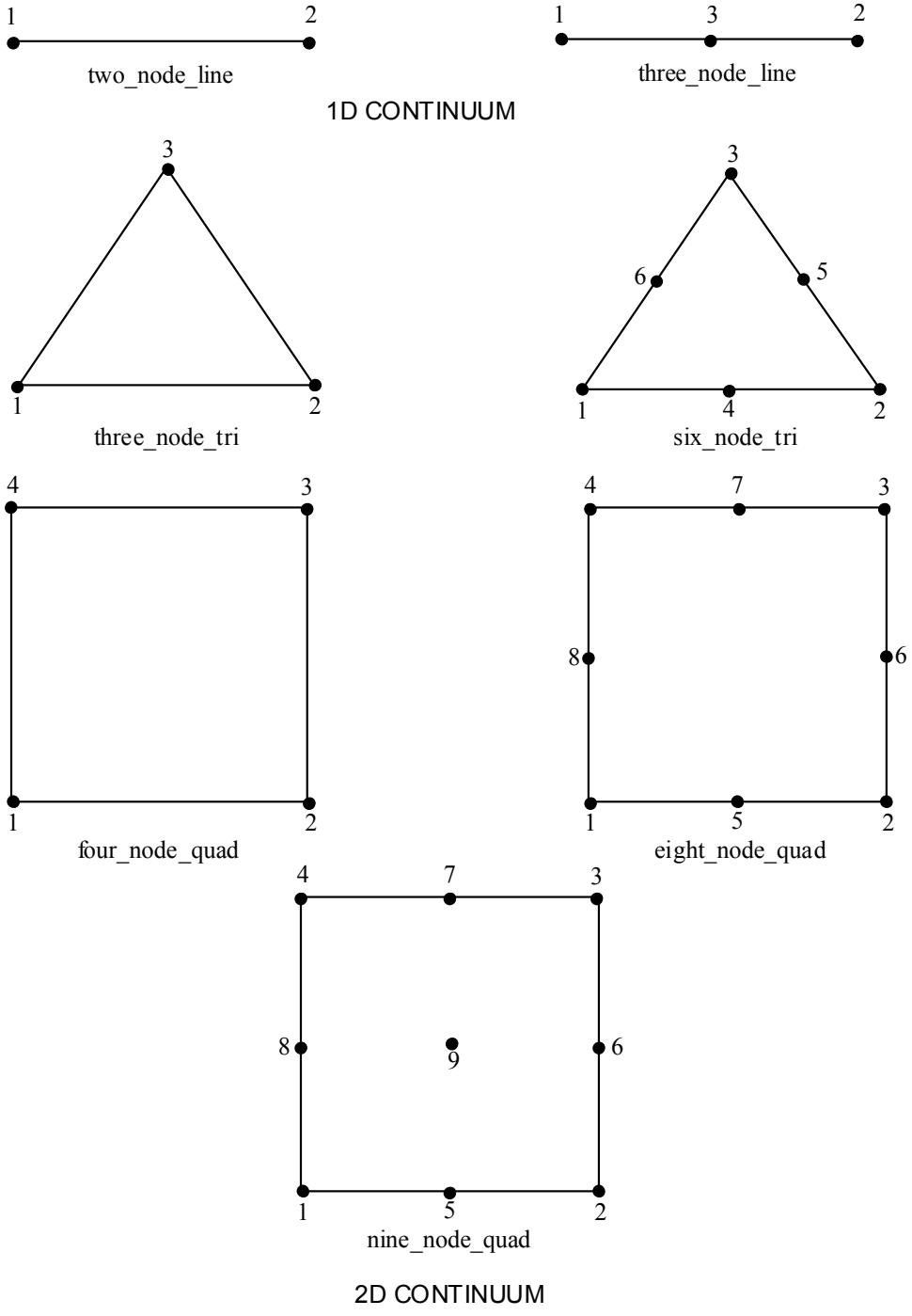
1 Darcy = 10^{-12} m^2

Appendix C

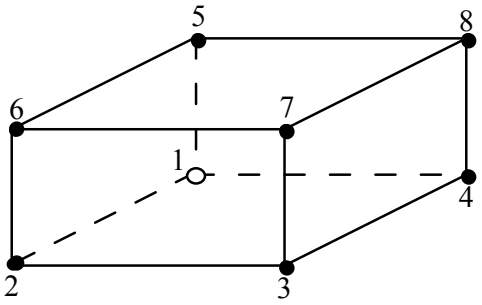
Quantity		Units / Conversion
General		
Acceleration		1 in/s^2 = 0.0254 m/s^2
Area		1 in^2 = 645.16 mm^2
Density	(i)	1 lbm/in^3 = 27679.905 kg/m^3
	(ii)	1 $slug/ft^3$ = 515.379 kg/m^3
Force		1 lb = 4.448 N (N = Newton)
Frequency		Hz (hertz = cycle/s)
Length		1 in = 0.0254 m; 1 ft = 0.3048 m
Mass	(i)	1 lbm = 0.45359 kg
	(ii)	1 slug = 14.594 kg
Moment		1 in-lb = 0.1130 N • m
Moment of inertia (area)		1 in^4 = 416231.4 mm^4
Moment of inertia (mass)	(i)	1 lbm-in^2 = 2.9264 10^{-4} kg • m^2
	(ii)	1 slug-in^2 = 0.009415 kg • m^2
Power	(i)	1 in-lb/s = 0.1130 W (watt = J/s)
	(ii)	1 hp = 0.746 kW (1 hp = 550 ft-lb)
Pressure		1 psi = 6894.8 Pa (psi = pounds/in^2; Pa = N/m^2)
Stiffness		1 lb/in = 175.1 N/m
Stress	(i)	1 psi = 6894.8 Pa
	(ii)	1 ksi = 6.8948 MPa; 1 MPa = 145.04 psi (ksi = 1000 psi; MPa = 10^6Pa)
Time		s (second)
Velocity		1 in/s = 0.0254 m/s
Volume		1 in^3 = 16.3871 $10^{-6}m^3$
Work, energy		1 in-lb = 0.1130 J (joule = N • m)
Heat Transfer		
Convection coefficient		1 Btu/h.ft^2 °F = 5.6783 W/m^2 °C
Heat		1 Btu = 1055.06 J (1 Btu = 778.17 ft-lb)
Heat flux		1 Btu/h.ft^2 = 3.1546 W/m^2
Specific heat		1 Btu/°F = 1899.108 J/°C
Temperature	(i)	T °F = [(9/5)T + 32] °C
	(ii)	T °K = T °C + 273.15 (K = kelvin)
Thermal conductivity		1 Btu/h.ft °F = 1.7307 W/m. °C
Fluid Flow		
Absolute viscosity		1 lb.s/ft^2 = 478.803 P (poise = g/cm • s)
Kinematic viscosity		1 ft^2/s = 929.03 St (stroke = cm^2/s)
Electric and Magnetic Fields		
Capacitance		F (farad)
Charge		C (coulomb)
Electric charge density		C/m^3
Electric potential		V (volt)
Inductance		H (henry)
Permeability		H/m
Permittivity		F/m
Scalar magnetic potential		A (ampere)

Appendix D

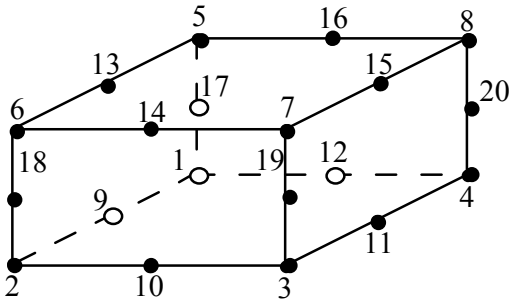
ELEMENT TYPES / SHAPES



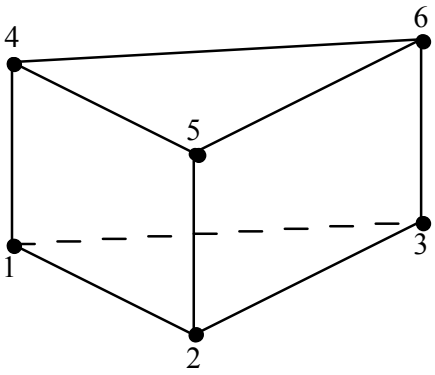
Appendix D



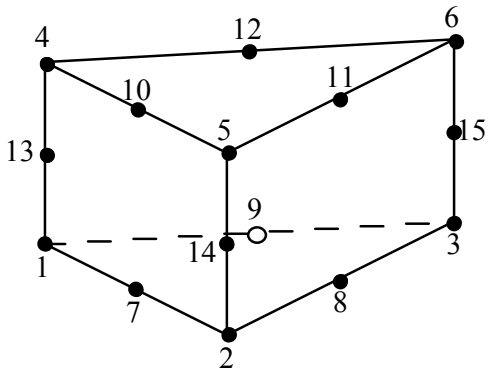
eight_node_brick



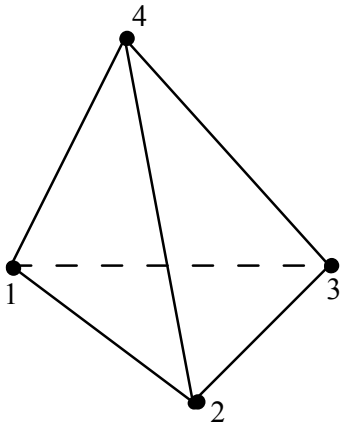
twenty_node_brick



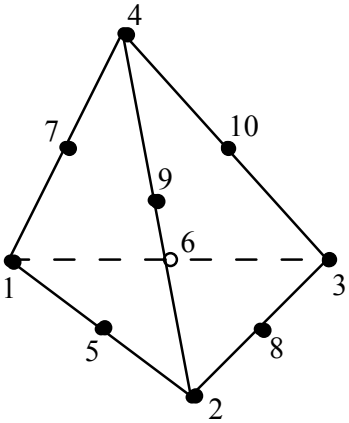
six_node_wedge



fifteen_node_wedge



four_node_tetra



ten_node_tetra

3D CONTINUUM

Appendix E

FORMAT OF DYNAFLOW OUTPUT FILES

This Appendix defines the format of files generated by DYNAFLOW. The data may then be read and converted into the form required by a post-processing program.

Each data set starts with a header of the form:

title

which contains up to 80 characters and is the same as the title defined by the command "DEFINE_PROBLEM" (see Sections 2.1 and 2.2).

Each output file is closed by the following data lines:

title
-1

E.1 TAPE90.name: Nodal Coordinates/Connectivity Data

```
title
numnp, nsd, ndof, numeg
    for node = 1, numnp
        node, (x (i, node), i=1, nsd)
    end
-1
    for neg = 1, numeg
        neg, idum, nen, numel, iopt, el_shape, el_name, reg_name
        for ne = 1, numel
            ne, mat(ne), (ien(i,ne), i=1, nen)
        end
    end
```

where:

numnp	=	number of nodal points
nsd	=	number of spatial dimensions
ndof	=	number of degrees of freedom per node
numeg	=	number of element groups
neg	=	element group number
nen	=	number of nodes per element
numel	=	number of elements
iopt	=	0 2d plane analysis
		1 1d analysis
		2 axisymmetric analysis
		3 3d analysis

el_shape = element shape (character string)
el_name = element name (character string)
reg_name = region name (character string)
ne = element number
mat = material number
ien = connectivity list

E.2 TAPE87.name: Nodal Results

```
title
ns, io, label, 'step', ns
    for node = 1, numnp
        node, (d(i, node), i=1, ndof)
    end
-1
```

where:

ns = step number
label = 'displacement' (io=1)
'velocity' (io=2)
'acceleration' (io=3)
'reaction' (io=4)
'eigenshape' (io=5)

E.3 TAPE96.name: Strain Energy

```
title
ns, io, 'field_w', 'step', ns
    for ne = 1, numel_tot
        ne, neg, energy
    end
-1
```

E.4 TAPE89.name: Field Results

```
title
ns, io, 'field comp', 'step', ns, 'ncomp', ncomp
    for neg = 1, numeg
        for ne = i, numel (neg)
            ne, (field (i,ne), i=1, ncomp)
        end
    end
-1
```

E.5 TAPE88.name: Time Histories

title
nts, 'time'
(time(i), i=0, nts) (8 values per line)
title
nts, i1, i2, io, neg, label1, label2, label3, i1, label4, neg
(comp(i), i=0, nts)

where:

nts	=	number of time steps
i1	=	node/element number
i2	=	component number
label1	=	'displ.' for diplacement (io=4)
		'veloc.' for velocity (io=5)
		'acc.' for acceleration (io=6)
		'react' for reaction (io=15)
		'stress' for solid element (io=1,2,3)
label2	=	component name (character string)
label3	=	'node' for nodal time history
		'elmnt' for field time history
label4	=	'group' for field time history
neg	=	group number (for field time history)

Appendix E

Notes . .

Index

Index

D

Index

Index

Q

R

Index

Z