FEM Manual

Version 1.1

Edition 0.1/14 February 2011

Farhat Research Group (FRG)
Stanford University

This manual was prepared with Texinfo (http://www.gnu.org/software/texinfo).

Short Contents

Table of Contents

Up: (dir)

FEM

Next: INSTALLATION

1 INTRODUCTION

This section compiles basic information on how to prepare an input data for a Finite Element analysis using the FEM code. The input data consists of command macros, clustered numerical data, and comment lines. Each data cluster is preceded by a command statement.

The following rules of thumb are suggested:

1 The users are allowed to have as many comment lines as they wish, as long as the first column of each line starts with

* (see COMMENTS).

2 The subsequent data clusters may appear in any order. The command statement preceding each cluster may be in lower, upper, or mixed characters.

3 The data in each cluster are free format.

4 **Only the first four letters of a command need to be specified unless otherwise noted.**

Any segment of a FEM input file — for example, the data corresponding to any command macro, or a command macro and its data, or any number of input lines — can be replaced by a statement of the form

```
INCLUDE        filename (or "filename")
```

where `filename` is the name of the file containing the information. "`filename`" can contain a path or can be replaced by `<filename>` in which case the path is that of the environment variable `\$FEM\_INCLUDE`. Furthermore, the included file `filename` can be compressed by gzip, bzip2, or zip.

The following subsections discuss in detail the syntax of each data cluster command.

Next: ATDARB, Previous: INTRODUCTION

2 INSTALLATION

The installation of **FEM** on a given computing system requires the availability on that system of the following tools:

`C++` compiler `g++`	Version 4.2.1 or higher.
`Fortran` compiler `gfortran`	Version 4.2.1 or higher.
`Flex` utility	Version 2.5 or higher. `Flex` is a lexical analyser required for building the parser of **FEM**'s input command data file.
`Bison` utility	Version 2.3 or higher. `Bison` is a parser generator required for building the parser of **FEM**'s input command data file.
`CMake` utility	Version 2.6 or higher. `CMake` is a cross-platform open-source build system. It is comparable to the Unix `Make` program in that the build process is ultimately controlled by configuration files (`CMakeLists.txt`). However unlike `Make`, it does not directly build the final software but instead generates standard build files such as makefiles for Unix and projects/workspaces for Windows Visual C++. The `CMake` version 2.6

utility can be obtained from http://www.cmake.org. (Note: a "README.cmake" file discussing details on `cmake` options for code configuration and installation is available in the directory containing the source code of **FEM**).

and following libraries:

`BLAS` library	`BLAS` is a set of Basic Linear Algebra Subprograms required by various operations performed in **FEM**.
`LAPACK` library	`LAPACK` is a high-performance Linear Algebra PACKage with advanced solvers.
`MPI` library `openmpi`	Version 1.2.6 or higher. `MPI` is a high-performance Message Passing Library required for performing interprocessor communication, among others. More specifically, **FEM** requires an MPI-2 implementation such as the one provided by the Open MPI project.
`OpenMP` API	Open Multi-Processing is an Application Programming Interface (API) that supports multi-platform shared memory multiprocessing programming in C, C++ and Fortran on many architectures, including Unix. As an option, **AERO-F** can be compiled with `OpenMP` to enable multi-threaded execution.

In addition, the `SPOOLES`, `MUMPS`, and `ARPACK` libraries are required when the user wishes to specify the `spooles` or `mumps` sparse solver, and/or the `ARPACK` eigen solver, respectively:

`SPOOLES` library	`SPOOLES` is a library for solving sparse real and complex linear systems of equations with a sparse direct solver, written in the C language using object oriented design.
`MUMPS` library	`MUMPS` is a library for solving sparse real and complex linear systems of equations with a multifrontal massively parallel sparse direct solver.
`ARPACK` library	`ARPACK` is the Arnoldi PACKage for the solution of large-scale symmetric, nonsymmetric, and generalized eigenproblems.

To install **FEM**, follow the procedure specified below:

- Edit the `CMakeLists.txt` file and include the file paths to all required and desired optional libraries. Typically, the `gfortran`, `BLAS`, `LAPACK`, `openmpi`, and `OpenMP` libraries will be automatically found. However, it may be necessary to specify the paths for the `SPOOLES`, `MUMPS`, and `ARPACK` libraries.
- From the directory containing the source code of **FEM**, type "`cmake -DAERO=1 .`" (without the " "). Note the space and the "." after `-DAERO=1`. The "." specifies the current directory.
- Watch the computer screen and verify that all invoked libraries were found and all build options were correct. A sample computer screen output of the command `cmake` is:

```
SEND ME WHAT TO PUT HERE
```

- Then, also from the directory containing the source code of **FEM**, type `make`.

The successful completion of the procedure described above leads to the creation in the bin/ directory of **FEM**'s executable fem.opt.

Next: ATDDIR, Previous: INSTALLATION

3 ACOUSTIC TIME-DOMAIN ARTIFICIAL BOUNDARY *S*

Command Statement:	**ATDARB**

The ATDARB command statement is used to specify the artificial boundary Σ on which an Antoine absorbing condition is to be applied in the time domain, and the order of this absorbing boundary condition. The input format is given below.

ATDARB	ORDER

FACE	FACE_TYPE	CONNECTIVITY_NODES

ORDER	Order of Antoine's absorbing boundary condition in the time domain (real). Currently supported values are 0 and 1.
FACE	Face (or edge in two dimensions) identification number whose type and connectivity are to be specified (integer). In practice, this identification number is ignored by **FEM**.
FACE_TYPE	
1	2-node line segment. To be used with two-dimensional linear elements.
2	3-node line segment. To be used with two-dimensional quadratic elements.
3	3-node triangular face. To be used with three-dimensional linear tetrahedral element.
4	4-node quad face. To be used with three-dimensional linear hexahedral element.
6	6-node triangular face. To be used with three-dimensional quadratic tetrahedral element.
10	n^2 -node arbitrarily higher-order quad face where n is the number of nodes on an edge of this face. To be used with element type 95.
11	arbitrarily higher-order triangular face.
12	arbitrarily higher-order line segment.
13	edge of a full isoparametric triangular element where the nodes are numbered linearly along it.
14	n^2 -node arbitrarily higher-order (spectral) quad face where n is the number of nodes on an edge of this face. To be used with spectral element type 105.
CONNECTIVITY_NODES	These should be listed in a stacked fashion on a single line, and numbered clockwise (when looking from infinity in three dimensions).

Next: ATDDNB, Previous: ATDARB

4 ACOUSTIC TIME-DOMAIN DIRICHLET BOUNDARY CONDITIONS *S*

Command Statement:	**ATDDIR**

The ATDDIR command statement is used to specify *nodal* Dirichlet boundary conditions for a time-domain acoustic scattering problem. The input format is given below.

```
ATDDIR
```

```
NODE#          VALUE
```

NODE#	Node number where the Dirichlet boundary condition is specified (integer).
VALUE	Value of the specified boundary condition (real).

Next: ATDNEU, Previous: ATDDIR

5 ACOUSTIC TIME-DOMAIN DISTRIBUTED NEUMANN BOUNDARY CONDITION *S*

Command Statement:	**ATDDNB**

The ATDDNB command statement is used to specify the surface of a scatterer on which a *distributed* Neumann boundary condition of the form $\frac{\partial p^s}{\partial n} = c$ is applied in a time-domain acoustic computation, and the value of the constant *c*.

The input format of this command is given below.

```
ATDDNB          CONSTANT
```

```
FACE          FACE_TYPE          CONNECTIVITY_NODES
```

CONSTANT	Value of the constant *c* (real).
FACE	Face (or edge in two dimensions) identification number whose type and connectivity are to be specified (integer). In practice, this identification number is ignored by **FEM**.
FACE_TYPE	
1	2-node line segment. To be used with two-dimensional linear elements.
2	3-node line segment. To be used with two-dimensional quadratic elements.
3	3-node triangular face. To be used with three-dimensional linear tetrahedral element.

4	4-node quad face. To be used with three-dimensional linear hexahedral element.
6	6-node triangular face. To be used with three-dimensional quadratic tetrahedral element.
10	n^2 -node arbitrarily higher-order quad face where n is the number of nodes on an edge of this face. To be used with element type 95.^M
11	arbitrarily higher-order triangular face.
12	arbitrarily higher-order line segment.
13	edge of a full isoparametric triangular element where the nodes are numbered linearly along it.
14	n^2 -node arbitrarily higher-order (spectral) quad face where n is the number of nodes on an edge of this face. To be used with spectral element type 105.
CONNECTIVITY_NODES	These should be listed in a stacked fashion on a single line, and numbered clockwise (when looking from infinity in three dimensions).

Next: ATDROB, Previous: ATDDNB

6 ACOUSTIC TIME-DOMAIN NEUMANN BOUNDARY CONDITIONS *S*

Command Statement:	**ATDNEU**

The ATDNEU command statement is used to specify the *nodal* Neumann boundary conditions for a time-domain acoustic scattering problem. The input format is given below.

```
ATDNEU
```

```
NODE#          VALUE
```

NODE#	Node number where the Neumann boundary condition is specified (integer).
VALUE	Value of the specified boundary condition (real).

Next: ACTUATORS, Previous: ATDNEU

7 ACOUSTIC TIME-DOMAIN ROBIN DISTRIBUTED BOUNDARY CONDITION *S*

Command Statement:	**ATDROB**

The ATDROB command statement can be used to specify the surface of a scatterer on which a *distributed* Robin boundary condition of the form $\alpha\frac{\partial p^s}{\partial n}+\beta p^s=\gamma$ is applied in a time-domain acoustic computation, and the three constants α, β, and γ.

The input format of this command is given below.

ATDROB	α	β	γ

FACE	FACE_TYPE	CONNECTIVITY_NODES

α	Non-zero constant premultiplying the normal derivative of the Robin boundary condition (real).
β	Constant premultiplying the unknown of the Robin boundary condition (float).
γ	Constant right hand-side of the Robin boundary condition (float).
FACE	Face (or edge in two dimensions) identification number whose type and connectivity are to be specified (integer). In practice, this identification number is ignored by **FEM**.
FACE_TYPE	
1	2-node line segment. To be used with two-dimensional linear elements.
2	3-node line segment. To be used with two-dimensional quadratic elements.
3	3-node triangular face. To be used with three-dimensional linear tetrahedral element.
4	4-node quad face. To be used with three-dimensional linear hexahedral element.
6	6-node triangular face. To be used with three-dimensional quadratic tetrahedral element.
10	n^2-node arbitrarily higher-order quad face where n is the number of nodes on an edge of this face. To be used with element type 95.^M
11	arbitrarily higher-order triangular face.
12	arbitrarily higher-order line segment.
13	edge of a full isoparametric triangular element where the nodes are numbered linearly along it.
14	n^2-node arbitrarily higher-order (spectral) quad face where n is the number of nodes on an edge of this face. To be used with spectral element type 105.

CONNECTIVITY_NODES	These should be listed in a stacked fashion on a single line, and numbered clockwise (when looking from infinity in three dimensions).

Next: AERO, Previous: ATDROB

8 ACTUATORS *S*

Command Statement:	**ACTUATORS**

The ACTUATORS command statement is used to specify time variant prescribed forces at any node and its associated dofs using a user defined subroutine, and the structural state at that node (displacement, velocity, and acceleration). The user has to write his/her own algorithm for specifying the prescribed forces within a subroutine named “control.C”. The SENSORS command statement must be used to pass to “control.C” the structural state at nodes of interest (see SENSORS). The user should also grab the special makefile for this command which is located within the FEM.d/Control.d directory, and use LOAD to activate this command (see LOAD). An example input file using this command can be found in FEM.d/fem_examples/Control.d/

Note 1: For nonlinear analyses, sensor information on the velocity and acceleration of a nodal degree of freedom is not currently available for use in the “control.C” file. These are currently passed as zero.

The syntax for invoking this option is given below.

```
ACTUATORS
```

```
NODE#          DOF#
```

NODE#	Node number where an actuating force is specified (integer).
DOF#	Degree of freedom local number where an actuating force is specified (integer).

Next: AEROH, Previous: ACTUATORS

9 AEROELASTICITY

Command Statement:	**AERO**

The AERO command statement can be used to perform any or all of the following instructions:

- indicate that **FEM** is to interact with **AERO-F** to compute a flow-induced load and perform the corresponding static structural analysis, or perform a coupled static or dynamic aeroelastic simulation

- choose a staggered time-integration algorithm (ALGORITHM subcommand) and, if needed, a displacement predictor (ALGORITHM

subcommand) defined by two coefficients α_0 and α_1 and the formula

$$u^{n+1_P} = u^n + \alpha_0 \Delta t_S \dot{u}^n + \alpha_1 \Delta t_S (\dot{u}^n - \dot{u}^{n-1})$$

when the structural time-integrator is implicit, or

$$u^{n+\frac{1}{2}_P} = u^n + \alpha_0 \Delta t_S \dot{u}^{n-\frac{1}{2}} + \alpha_1 \Delta t_S (\dot{u}^{n-\frac{1}{2}} - \dot{u}^{n-\frac{3}{2}})$$

when the structural time-integrator is explicit, to perform a static or dynamic aeroelastic simulation.

Here, u^{n+1_P} ($u^{n+\frac{1}{2}_P}$) is the predicted displacement field at time-step t^{n+1} ($t^{n+\frac{1}{2}}$).

- select an algorithm for computing the corrected pressure field P_c^{n+1} to be used in computing the aerodynamic forces acting on the structure at time-step t^{n+1} (PRESSURE subcommand) of a static or dynamic aeroelastic simulation

The purpose of the displacement predictor and that of the force corrector are to compensate for the effects of time-lagging. More specifically, they improve the time-accuracy and numerical stability of the chosen staggered procedure by minimizing the lack of energy conservation at the fluid-structure interface. The theoretical, algorithmic, and practical aspects of both of these "compensators" are described in S. Piperno and C. Farhat, "Partitioned Procedures for the Transient Solution of Coupled Aeroelastic Problems - Part II: Energy Transfer Analysis and Three-Dimensional Applications," Computer Methods in Applied Mechanics and Engineering, Vol. 190, pp. 3147-3170 (2001).

Before setting the values of α_0 and α_1 that complete the definition of the displacement predictor and choosing the force corrector, the user should note that:

- Currently, the fluid code **AERO-F** can send to **FEM** at t^{n+1} either the pressure field at t^{n+1} , P^{n+1} , or the averaged pressure field $\overline{P}^{n+1} = \frac{1}{\Delta t_S} \int_{t^n}^{t^{n+1}} P(t) dt$. Hence, P_c^{n+1} can be set at least to either of these two values. Therefore, the purpose of the PRESSURE subcommand is to propose additional choices for the corrected pressure field P_c^{n+1} . These can affect the order of time-accuracy of the chosen staggered solution procedure (see C. Farhat, G. van der Zee and P. Geuzaine, "Provably Second-Order Time-Accurate Loosely-Coupled Solution Algorithms for Transient Nonlinear Computational Aeroelasticity," Computer Methods in Applied Mechanics and Engineering, Vol. 195, pp. 1973-2001 (2006)).
- The values of α_0 and α_1 that minimize the variation of energy exchanged at the fluid-structure interface and therefore improve time-accuracy and numerical stability of the chosen staggered solution procedure depend on the chosen fluid and structure time-integrators, the chosen force corrector, and the selected staggered solution procedure itself.
- When the Generalized α method is used for time-integrating linear or nonlinear structural dynamics problems (see DYNAMICS), **FEM** solves the following equilibrium problem

$$M\ddot{u}^{n+1-\alpha_m} + C\dot{u}^{n+1-\alpha_f} + F_{internal}(u^{n+1-\alpha_f}) = F_{external}^{n+1-\alpha_f}$$

- When the central difference method is used for time-integrating structural dynamics problems (see DYNAMICS), **FEM** solves the following equilibrium problem

$$M\ddot{u}^{n+1} + C\dot{u}^{n+1} + F_{internal}(u^{n+1}) = F_{external}^{n+1}$$

Note 1: AERO-F offers two different computational frameworks for fluid-structure interaction: an Arbitrary Lagrangian/Eulerian (ALE) framework, and an embedded one. When a functional capability described below is meaningful, applicable, or supported for only one of these two computational frameworks, its designating keyword is followed by *[ALE]* in the case of the ALE computational framework, and by *[EMB]* in the case of the embedded computational framework. When this capability is applicable, meaningful, and supported for both computational frameworks, its designating keyword is not followed by any symbol.

Note 2: The PP, MPP, and C0 algorithms require the additional presence of the DYNAMICS command in the input file. The A0, A4, A5, and A6 algorithms require the additional presence in the input file of the QSTATICS command for a static aeroelastic analysis, and that of the DYNAMICS command for a dynamic aeroelastic analysis. The B0 algorithm requires the additional presence in the input file of the QSTATICS command for a static aeroelastic analysis.

Note 3: The values of α_0 and α_1 derived in the paper by Piperno and Farhat mentioned above correspond to the case where the structure is linear and time-integrated by the Newmark algorithm with $\beta = \frac{1}{4}$ and $\gamma = \frac{1}{2}$ (see DYNAMICS). For different configurations of the structural time-integrator, that analysis needs to be redone if it is to be used for choosing the parameters of the AERO command.

Note 3: If an aeroelastic analysis is requested with the GEPS and IDISP6 commands present in the input file, and the IDISPLACEMENTS command is specified to initialize the displacement field, then **FEM** sends to the fluid code at each time step the sum of the updated displacement field (interpreted in that case as an increment displacement) and the displacement specified under the IDISP6 command. If in such a case the IDISPLACEMENTS command is not present in the input file, the displacement field of the structure is then initialized by the IDISP6 command, and **FEM** acts as follows: at the first time-step, it sends to the fluid code only the updated displacement field (since initialization accounts in that case for the content of the IDISP6 command), but at each subsequent time-step, it sends the sum of the updated displacement field and the displacement specified under the IDISP6 command.

The syntax for invoking this option is given below.

AERO

ALGORITHM α_0 α_1
PRESSURE
MATCHER pathandfilename
EMBEDDED embeddedsurfaceid

ALGORITHM

PP *[ALE]* This "Ping-Pong" algorithm sends the initial displacement of the structure, specified under either the IDISP6 command or the IDISPLACEMENTS with the MODAL option command (see IDISPLACEMENTS), to the fluid code which receives it and deforms the fluid mesh accordingly. For this purpose, the DYNAMICS command must also be present in the **FEM** input file. After the send occurs, the structure code exits gracefully, the fluid code computes

the fluid mesh deformation associated with the specified structural displacement and outputs (if requested) the corresponding position and/or displacement of the fluid mesh (characters).

MPP *[ALE]* This "Multi-Ping-Pong" algorithm sends several initial displacements of the structure at a time — for example, modal displacements — to the fluid code, which uses them to compute compatible fluid mesh deformations. If a coefficient α_0 is specified after MPP, the initial displacements are amplified by α_0 before they are sent to the fluid code. However, the fluid code scales back by $1/\alpha_0$ the corresponding fluid mesh displacements it computes before saving them in an output file, in order to preserve the effect of the mass normalization of the modal structural displacements. The deformed fluid mesh configurations can then be used to generate sources of excitations for the construction of fluid POD bases and ROMs, or for linearized flow simulations whose initial conditions involve a (modal) position or velocity of the fluid surface mesh. The READMODE command must be used to input the initial displacements of interest — for example, a set of eigenmodes. As for the "Ping-Pong" case, the DYNAMICS command must also be present in the **FEM** input file, **FEM** exits gracefully after the send occurs, the fluid code computes the fluid mesh deformation associated with the specified structural displacements and outputs (if requested) the corresponding positions and/or displacements of the fluid mesh. In addition, for each inputted deformed fluid mesh position, the fluid code outputs an identification tag — for example, the frequency of the corresponding input structural mode in the case of modal displacements (characters).

A0 This is the basic sequential staggered solution procedure (characters). When subcycling is effected, the same algorithm has been referred to in the AIAA Paper 96-1388 by Farhat and co-workers as the A1 algorithm (characters). It should not be used without a displacement predictor as it would reduce the overall order of time-accuracy compared to that intrinsic to the structural time-integrator. For the same reason, it is also not recommended — and as a matter of fact not available — when the structural time-integrator is explicit.

A4 This is the basic staggered solution procedure with fluid-structure inter-parallelism (characters). It has been referred to as A2 in the AIAA Paper 96-1388 by Farhat and co-workers. When this algorithm is specified, **AERO-F** sends to **FEM** P^n or $\overline{P}^n$ rather than the corresponding pressure fields at t^{n+1}. Currently, this algorithm is not available when an explicit structural time-integrator is chosen.

A5 This staggered solution procedure also features inter-parallelism but offers a better accuracy than A4 when both methods are used without a displacement predictor (characters). Currently, this algorithm is not available when an explicit structural time-integrator is chosen.

A6 This staggered solution procedure has superior stability and accuracy properties. It is recommended when the fluid is time-advanced by an implicit time-integrator. It has been referred to in most papers by Farhat and co-workers as the ISS (Improved Sequential Staggered algorithm) method (characters). Originally, it was thought that this staggered solution procedure does not require any displacement prediction ($\alpha_0 = \alpha_1 = 0$) because it incorporates its own first-order prediction. However, it was later discovered that this partitioned procedure is genuinely second-order time-accurate if it is equipped with a second-order displacement prediction. In that case, α_0 can be set to $\alpha_0 = 0$ in **FEM** because it is reset in any case by **AERO-F** to the appropriate value ($\alpha_0 = 1/2$) and α_1 should be set to $\alpha_1 = 1/8$. Also, this partitioned solution procedure prefers a non time-averaged pressure field. Most importantly, it was designed to be used exclusively with the midpoint implementation of the Newmark time-integrator with $\beta = \frac{1}{4}$ and $\gamma = \frac{1}{2}$ (see DYNAMICS), and with the NONCOLLOCATED scheme of aerodynamic force evaluation (see below) (characters). Currently, this algorithm is not available when an explicit structural time-integrator is chosen.

BO This algorithm sends the initial displacment of the structure (if any) to the flow solver. After receiving it and updating its mesh, **AERO-F** performs any requested flow simulation and sends to **FEM** upon completion the flow-induced load. Then, **FEM** receives this load, performs a static analysis if requested, and finally outputs any requested data (such as the computed flow-induced load, and/or resulting displacement, stress, and strain fields).

CO This staggered solution procedure is designed for the case where the structural subsystem is time-integrated by the explicit central difference method (see DYNAMICS), and the fluid subsystem is time-integrated by either an explicit or an implicit scheme. As shown in C. Farhat, A. Rallu, K. Wang and T. Belytschko, "Robust and Provably Second-Order Explicit-Explicit and Implicit-Explicit Staggered Time-Integrators for Highly Nonlinear Fluid-Structure Interaction Problems," International Journal for Numerical Methods in Engineering, (2010), it is genuinely second-order time-accurate when equipped with the second-order displacement predictor obtained by setting $\alpha_0 = 0.5$ and $\alpha_1 = 0.375$ (see below), the second-order force corrector obtained by setting PRESSURE to COLLOCATED (see below), and a second (or higher)-order accurate explicit or implicit ALE fluid time-integrator.

α_0 When used with the MPP command, this coefficient can be specified to amplify the initial displacements to be sent to the fluid code (however, as noted above, the fluid mesh displacements computed by the fluid code are scaled back by $1/\alpha_0$ before they are saved in an output file). In this case, the default value is 1. Otherwise, this coefficient is part of the construction of a predictor for the position of the structure, in which case the default value is 0. For $\alpha_0 = 0$ (and $\alpha_1 = 0$), no prediction is effected. Setting $\alpha_0 = 1$ (and $\alpha_1 = 0$) generates a first-order prediction, and setting $\alpha_0 = 1$ (and $\alpha_1 = \frac{1}{2}$) generates a second-order prediction. However, the optimal values of α_0 and α_1 depend on the specifics of the fluid and structure field time-integrators (see the aforementioned paper on interface energy conservation by Piperno and Farhat) (real).

α_1 See above (real).

PRESSURE

COLLOCATED In this case, P_c^{n+1} is set to the received pressure field P^{n+1} (or $\overline{P}^{n+1}$), then converted to $P_c^{n+1-\alpha_f} = (1-\alpha_f)P_c^{n+1} + \alpha_f P_c^n$ before the aerodynamic forces are constructed and fed into the structural equations of dynamic equilibrium (see the introduction to this command). Therefore, this option covers the cases $P_c^{n+1} = P^{n+1}$ and $P_c^{n+1} = \overline{P}^{n+1}$, which explains the origin of the word "collocated" (characters).

NONCOLLOCATED In this case, which is also the default value of PRESSURE, the pressure field P_c^{n+1} is set to $P_c^{n+1} = \frac{1}{\gamma}P^{n+1} - \frac{(1-\gamma)}{\gamma}P_c^n$ (or $P_c^{n+1} = \frac{1}{\gamma}\overline{P}^{n+1} - \frac{(1-\gamma)}{\gamma}P_c^n$), then converted to $P_c^{n+1-\alpha_f} = (1-\alpha_f)P_c^{n+1} + \alpha_f P_c^n$ (here, γ is the γ parameter of the Generalized α method). This specific choice for P_c^{n+1} is consistent with the quadrature rule of the Generalized α method for evaluating the work done by an external force in the time-interval $[t^n,\ t^{n+1}]$. In other words, the pressure forces fed into the structural equations of equilibrium are based on $P_c^{n+1-\alpha_f} = \frac{(1-\alpha_f)}{\gamma}P^{n+1} + (1 - \frac{(1-\alpha_f)}{\gamma})P_c^n$. For

$\gamma = \alpha_f = \frac{1}{2}$, $P_c^{n+1-\alpha_f} = P^{n+1}$, which explains the origin of the word “non collocated” (characters).

MATCHER

Name (including path, if needed) of the structural matcher file (characters).

pathandfilename

EMBEDDED

embeddedsurfaceid

Integer identification number of a surface defined in SURFACETOPO using 3-noded triangles (integer). This surface, which is to be embedded in an **AERO-F** CFD mesh for fluid-structure computations, is assumed to be identical to the wet surface of the finite element structural model and defined using the same nodes of this model. For this reason, when the EMBEDDED option is used, it is not currently necessary to provide a structure matcher file.

Next: ATTRIBUTES, Previous: AERO

10 AEROHEAT (FEM-HEAT Only)

Command Statement:	**AEROH**

The AEROH command statement is used to indicate that FEM-HEAT is to interact with a fluid code to perform an aerothermal (thermostructure-thermofluid) coupled simulation, to select a staggered time-integration algorithm (ALGORITHM subcommand), and specify the α_0 and α_1 parameters of the following prediction of the temperature of the structure at time-step t^{n+1} (ALGORITHM subcommand)

$$T_S^{n+1^p} = T_S^n + \alpha_0 \Delta t_S^H \dot{T_S}^n + \alpha_1 \Delta t_S^H (\dot{T_S}^n - \dot{T_S}^{n-1})$$

The syntax for invoking this option is given below.

AEROH

ALGORITHM	α_0	α_1

ALGORITHM

A0 This is the basic sequential partitioned procedure. It is similar to the A0 algorithm of the AERO command.

α_0 For $\alpha_0 = 0$ (and $\alpha_1 = 0$), no prediction is effected. Setting $\alpha_0 = 1$ (and $\alpha_1 = 0$) generates a first-order prediction,and setting $\alpha_0 = 1$ (and $\alpha_1 = \frac{1}{2}$) generates a second-order prediction. However, the optimal values of α_0 and α_1 depend on the specifics of the fluid and structure field time-integrators (see the

CAS paper on energy conservation by Piperno and Farhat) (real).

See above (real).

Next: BOFFSET, Previous: AEROH

11 ATTRIBUTES *S*

Command Statement:	**ATTRIBUTES**

The ATTRIBUTES command statement is used to label an element with an attribute identification number for linear material properties and/or geometric properties, a composite or orthotropic shell, or anisotropic solid element attribute identification number, and a frame or fiber-angle attribute identification number if this element has been defined as a composite or orthotropic shell or anisotropic solid element.

The attribute identification number for linear material properties and/or geometric properties is used in the MATERIAL command when specifying the linear material properties and/or geometric properties of a group of elements (see MATERIAL).

An “empty” (or “phantom”) element — that is, an element with zero generalized stiffness and mass matrices — can be useful in aeroelastic, aerothermal, and aerothermoelastic computations to facilitate the exchange of elastodynamic, thermal, and aerodynamic data between the fluid, thermal, and structural analyzers. Such an element can be specified by assigning it a negative attribute number. In this case, it is not necessary to specify a material for such an element as its properties will be ignored.

An element that is not an “empty” (or “phantom”) element and which does not require any material property should be assigned zero (0) as an attribute number and no material property. Examples of such an element are rigid elements and frequency-domain acoustic elements in a homogeneous medium.

On the other hand, an element that is not assigned an attribute number is simply ignored by **FEM**.

FEM supports three input formats for this command that are described below. In the case of the first format, there should be as many lines as the number of finite elements. All formats can be mixed.

Note 1: When “empty” (or “phantom” elements) are used in the mesh, the FETI solvers are not guaranteed to work because of issues related to subdomain singularities. Hence, it is not recommended to use FETI solvers in such cases.

```
ATTRIBUTES
```

```
ELEMENT#        MAT_ATT#        CMP_ATT#        CMP_FRM#
```

or

ELEMENT#	MAT_ATT#	CMP_ATT#	THETA	θ_{Ref}

or

STARTING_ELEMENT#	ENDING_ELEMENT#	MAT_ATT#	CMP_ATT#	CMP_FRM#

or

STARTING_ELEMENT#	ENDING_ELEMENT#	MAT_ATT#	CMP_ATT#	THETA	θ_{Ref}

or

STARTING_ELEMENT#	ENDING_ELEMENT#	IDENTITY

ELEMENT# Element number whose attribute numbers are to be specified (integer).

MAT_ATT# Group identification number (integer); **FEM** supports gaps in numbering. Identifies material properties.

CMP_ATT# Composite (orthotropic or anisotropic) identification number (integer); **FEM** supports gaps in numbering. Identifies composite (orthotropic or anisotropic) properties. May be left blank if element is not a composite (orthotropic or anisotropic) element.

CMP_FRM# Composite frame identification number (integer); **FEM** supports gaps in numbering. May be left blank if element is not a composite (orthotropic or anisotropic) element.

THETA Keyword that must be spelled as THETA (characters). This keyword announces that the next item on the same input line is the value θ_{Ref} of THETA. This unusual **FEM** input is necessary to avoid parsing conflicts.

θ_{Ref} Reference angle in degrees between the axis defined by local nodes 1 and 2 of the element, and the material local x-axis defining the direction of the σ_{xx} and ϵ_{xx} of the constitutive law (real). When also using the LAYN and LAYC commands, the material x-axis corresponds to the direction of $E_1^{(k)}$ when $\theta_F^{(k)}$ is zero. If this angle rather than CMP_FRM# is specified, the composite frame is internally generated by **FEM**.

STARTING_ELEMENT# First element of a sequence of elements that have the same MAT_ATT#, CMP_ATT#, and CMP_FRM# (integer).

ENDING_ELEMENT# Last element of a sequence of elements that have the same MAT_ATT#, CMP_ATT#, and CMP_FRM# (integer).

MAT_ATT#	Group identification number. Identifies material properties (integer).
CMP_ATT#	Composite identification number (integer). Identifies composite (orthotropic or anisotropic) properties. May be left blank if element is not a composite (orthotropic or anisotropic) element.
CMP_FRM#	Composite frame identification number (integer). May be left blank if element is not a composite (orthotropic or anisotropic) element.
IDENTITY	This keyword signals to the **FEM** code that each element in the range delimited by STARTING_ELEMENT# and ENDING_ELEMENT# has a material attribute identification number equal to its element identification number. For example, the sequence 1 3 IDENTITY means that element#1 has the material attribute identification 1, element#2 has the material attribute identification 2, and element#3 has the material attribute identification 3.

Next: EFRAMES, Previous: ATTRIBUTES

12 BEAM OFFSET

Command Statement:	**BOFFSET**

The BOFFSET command statement is used to specify a beam's neutral axis offset from the line passing through its two end nodes. This command may be used for both Euler-Bernoulli and Timoshenko beam elements. For beam elements not listed under BOFFSET, a zero offset is used. If the BOFFSET command is absent altogether, then an offset of zero is used for all beam elements. If a listed element is neither an Euler-Bernoulli nor a Timoshenko beam element, then the offset is ignored for that element. The input format is as follows.

```
BOFFSET
```

```
STARTING_ELEMENT#    ENDING_ELEMENT#    x    y    z
```

STARTING_ELEMENT#	First element of a sequence of elements that have the same offset specified by {x, y, z} (integer).
ENDING_ELEMENT#	Last element of a sequence of elements that have the same offset specified by {x, y, z} (integer). If ENDING_ELEMENT# is the same as STARTING_ELEMENT#, then the specified offset is applied to that element only (integer).
x	x component of the offset vector expressed in the global frame (float).
y	y component of the offset vector expressed in the global frame (float).
z	z component of the offset vector expressed in the global frame (float).

Next: BINARY, Previous: BOFFSET

13 BEAM REFERENCE FRAMES

Command Statement:	**EFRAMES**

The EFRAMES command statement is used to specify a beam's orientation which defines the position of the beam with respect to the global frame. The requirements for each beam frame are that one of the axis, **the local x-axis**, concur with the longitudinal axis of the beam, and the remaining two axes complete an orthogonal triad. If this requirement is violated, FEM regenerates the local x-axis as well as the local z-axis.

FEM also supports a run-time generation of beam frames that is activated either when the target beam element is identified under this command and a third node is specified to generate the frame, or when a third node is found in the definition of a beam element within the TOPOLOGY command. The only requirement for the third node is that it does not be colinear with the other two beam nodes that define the local x-axis (X). Under *this* command, the third node defines the X,Y plane. Under the TOPOLOGY command, it defines the X,Z plane. Using the third node option under the TOPOLOGY command also relieves the user from specifying the EFRAMES command. An example illustrating the third node option in the TOPOLOGY command can be found in FEM.d/fem_examples/Third_Node.d

Otherwise, the input format for this command is given below with the number of lines equal to the number of beams in the problem.

```
EFRAMES
```

ELEMENT#	S_{1x} S_{1y} S_{1z}	S_{2x} S_{2y} S_{2z}	S_{3x} S_{3y} S_{3z}

or

```
ELEMENT#        THIRDNODE        third_node
```

ELEMENT# Element number where the beam frame is specified (integer).

S_{1x} S_{1y} S_{1z} The first axis, **the local x-axis**, of the beam frame expressed in the global frame. This axis must concur with the longitudinal axis of the beam (floats).

S_{2x} S_{2y} S_{2z} The second axis of the beam frame expressed in the global frame (floats).

S_{3x} S_{3y} S_{3z} The third axis of the beam frame expressed in the global frame (floats).

THIRDNODE Keyword that must be spelled as THIRDNODE. This keyword announces that the next item on the same input line is the value third_node# of THIRDNODE. This unusual FEM input is necessary to avoid parsing conflicts (characters).

THIRD_NODE# Id number of an existing node to be considered as a third node of the beam element identified by ELEMENT#, and which must be in the local x-y plane of the beam. The only requirement for the third node is that it does not be colinear with the other two beam nodes. The normalized vector node1 to node2 defines the local x axis. The normalized vector node1 to node3 defines the local y-axis. The local z-axis is automatically generated as cross product of the other two normalized axes.

Note 1: The above axes have to be normalized. By default, the beam frames concur with the global frame.

Next: CONVECTION, Previous: EFRAMES

14 BINARY INPUT / OUTPUT

Command Statement:	**BINARY**	[<pathandfileprefix>]

The command BINARY is used to instruct **FEM** to read a set of binary files containing distributed intput data associated with a mesh decomposition, and/or generate the specified output results (see OUTPUT) in similar binary distributed output files. Currently, the binary distributed input data files can be created by **SOWER** only. The purpose of **SOWER** is to generate distributed input files for distributed parallel codes, and merge their distributed output files into a single output file. Collectively, the binary distributed input files contain: (a) a distributed geometry and corresponding data associated with the commands of this User's Reference Manual marked by "*S*", (b) a distributed decomposition containing the mesh partition (optional, see **SOWER**'s User's Reference Manual), (c) a global file containing the cluster-to-subdomain and subdomain-to-subdomain connectivities (optional, see **SOWER**'s User's Reference Manual), and (d) a distributed matcher file (optional, see the **MATCHER** and **SOWER** User's Reference Manuals) for aeroelastic computations.

Note 1: This command is active only when **FEM** is compiled in the distributed memory execution mode where parallelism is implemented using a combination of the MPI and OpenMP parallel constructs (see SERIAL AND PARALLEL EXECUTIONS).

Note 2: As stated above, any command documented in this User's Reference Manual and marked by "*S*" can be assumed to be contained in the binary file described in (a) above. However, if that command contains both an algorithmic parameter as well as data, only the data will be contained in the appropriate binary file. Therefore, the command itself should be kept in the final **FEM** input file together with the algorithmic parameter, but without the data. For example, the command IDISPLACEMENTS specifies both an amplification factor and an initial displacement data. The initial displacement data will be included in the binary file mentioned above, but the amplification factor will not. This is to allow the user to change simple things like the amplification factor without having to re-run **SOWER** to produce the binary distributed input files.

The syntax for this command is as follows.

```
BINARY     <pathandfileprefix>
```

```
BINARYINPUT  flagBIN
BINARYOUTPUT flagBOUT
```

`pathandfileprefix` If instructed to read a set of binary input files containing distributed input data associated with a mesh decomposition, **FEM** expects the names of these files to be by default (`decomposition`, `connectivity`, `subdomains`, and `clusterData`. Alternatively, `pathandfileprefix` can be used to specify a prefix for these four files (and a path, if needed), in which case the expected name of the binary decomposition file is `fileprefix.dec`, that of the binary connectivity file is `fileprefix.con`, that of the binary subdomains file is `fileprefix.sub`, and that of the binary

	clusterData file is `prefixfile.msh` (characthers).
`BINARYINPUT`	Sub-command keyword to specify whether **FEM** is to read a specified set of binary input files containing distributed input data.
`flagIN`	On/Off flag (characters).
`on`	In this case, which is also the default case, **FEM** is to read a specified set of binary input files containing distributed input data.
`off`	In this case, **FEM** is not to read a specified set of binary input files containing distributed input data.
`BINARYOUTPUT`	Sub-command keyword to specify whether **FEM** is to generate the specified output results (see OUTPUT) in binary distributed output files. output files
`flagOUT`	On/Off flag (characters).
`on`	In this case, which is also the default case, **FEM** is to generate the specified output results (see OUTPUT) in binary distributed output files.
`off`	In this case, **FEM** is not to generate the specified output results (see OUTPUT) in binary distributed output files.

Next: DISPLACEMENTS, Previous: BINARY

15 BOUNDARY CONVECTION *S*

Command Statement:	**CONVECTION**

The `CONVECTION` command statement is used to specify nodal convection type boundary conditions. Each node can have only one degree of freedom. This command statement can be used to solve a prescribed boundary convection problem for both statics and dynamics. The input format is given below.

Note 1: This command contributes to the construction of the right-hand side vector only. For the left-hand side (stiffness matrix) contribution, FEM uses boundary convection elements and information specified in the `MATERIAL` command.

```
CONVECTION
```

```
NODE#        H-COEFF        AREA        TA
```

`NODE#`	Node number where the convection is specified (integer).
`H_COEFF`	Convection coefficient at the prescribed node (float).
`AREA`	Value of the node cross-sectional area (float).
`TA`	Ambient temperature around the node (float).

Next: FLUX, Previous: CONVECTION

16 BOUNDARY DISPLACEMENTS *S*

Command Statement:	**DISPLACEMENTS**

The DISPLACEMENTS command is used to prescribe nodal displacements and/or rotations, either directly, or via the definition of a surface using the command SURFACETOPO. In the latter case, the specified displacement or rotation is applied to each specified local degree of freedom of each node of that surface. The user can specify up to three displacements per node if a node can have up to three degrees of freedom, and up to three displacements and three rotations if it can have up to six degrees of freedom.

Note 1: This command can also be used to specify nodal Dirichlet boundary conditions for a time-domain of frequency-domain acoustic simulation by setting DOF# to 8 (see below). Hence, it can also serve as an alternative to the ATDDIR and HDIR commands.

The following two formats are available for this command and can be mixed.

```
DISPLACEMENTS
```

```
NODE#          DOF#          VALUE
```

```
SURFACE        SURFACE#      DOF#          VALUE
```

NODE#	Node number where the displacement or rotation is specified (integer).
DOF#	Degree of freedom local number where the displacement or rotation is specified (integer).
VALUE	Value of the specified displacement or rotation (real).
SURFACE	Keyword indicating that a surface defined in SURFACETOPO is to be identified next by its integer identification number (characters).
SURFACE#	Integer identification of the surface defined in SURFACETOPO where the force VALUE is specified (integer).

Next: FORCES, Previous: DISPLACEMENTS

17 BOUNDARY FLUXES

Command Statement:	**FLUX**

The FLUX command statement is used to specify nodal temperature flux type boundary conditions. Each node can have only one degree of freedom. This command statement can be used to solve a prescribed temperature flux boundary problem for both statics and dynamics. The input format is given below.

```
FLUX
```

```
NODE#          VALUE
```

NODE#	Node number where the flux is specified (integer).
VALUE	Value of the prescribed boundary flux (float).

Next: TEMPERATURES, Previous: FLUX

18 BOUNDARY FORCES *S*

Command Statement:	**FORCES**

The FORCES command is used to prescribe external forces and/or moments at the node level, either directly, or via the definition of a surface using the command SURFACETOPO. In the latter case, the specified force or moment is applied to each specified local degree of freedom of each node of that surface. The user can specify up to three forces per node if a node can have up to three degrees of freedom, and up to three forces and three moments if it can have up to six degrees of freedom. The user can also time-vary the specified forces and/or moments using the MFTT command.

Note 1: This command can also be used to specify nodal Neumann boundary conditions for a time-domain acoustic simulation by setting DOF# to 8 (see below). Hence, it can also serve as an alternative to the command ATDNEU (see ATDNEU).

The following two formats are available for this command and can be mixed.

```
FORCES
```

```
NODE#          DOF#          VALUE
```

```
SURFACE        SURFACE#      DOF#          VALUE
```

NODE#	Node number where the force or moment is specified (integer).
DOF#	Degree of freedom local number where the force or rotation is specified (integer).
VALUE	Value of the specified force or moment (real).
SURFACE	Keyword indicating that a surface defined in SURFACETOPO is to be identified next by its integer identification number (characters).
SURFACE#	Integer identification of the surface defined in SURFACETOPO where the force or moment VALUE is specified (integer).

Next: BUCKLE, Previous: FORCES

19 BOUNDARY TEMPERATURES *S*

Command Statement:	**TEMPERATURES**

The TEMPERATURES command statement is used to specify prescribed nodal temperature type boundary conditions. Each node can have only one degree of freedom. For structural analysis using FEM, this command statement causes FEM to construct a thermal load based on the prescribed nodal temperatures and the reference temperatures (see MATLAW, see TOPOLOGY, and use this load as usual in a structural analysis).

The input format is given below.

```
TEMPERATURES
```

```
NODE#          VALUE
```

NODE#	Node number where the temperature is specified (integer).
VALUE	Value of the prescribed temperature (float).

Next: COMMENTS, Previous: TEMPERATURES

20 BUCKLING

Command Statement:	**BUCKLE**

The BUCKLE command statement is to be used together with the GEPS, IDISP6 (with $\alpha = -1$), and EIGEN command statements to request the buckling analysis of a given structure.

When the EIGEN, GEPS, and IDISP6 (with $\alpha = -1$) command statements are specified in the input file, the BUCKLE command statement signals to FEM that the eigenvalue problem to be solved is $KU = -\lambda K_G U$, where K_G is the geometric stiffness matrix associated with a displacement field specified under the IDISP6 command statement, and whose computation is triggered by the presence of the GEPS command statement in the input file. For each eigenvalue λ , the corresponding buckling load is λf where *f* is the load that created the displacement field specified under the IDISP6 command.

An example input file that illustrates a buckling analysis can be found in FEM.d/fem_examples/Buckle.d The syntax for invoking this option is given below.

```
BUCKLE
```

Next: COMPOSITE, Previous: BUCKLE

21 COMMENTS

The '*' at the beginning of a line indicates that the subsequent input on the same line is a comment. For FEM, it can be placed anywhere.

Next: CFRAMES, Previous: COMMENTS

22 COMPOSITE (OR ORTHOTROPIC SHELL OR ANISOTROPIC SOLID)

Command Statement:	**COMPOSITE**

The COMPOSITE command statement is used to specify the material properties of a composite shell, or orthotropic shell, or anisotropic solid element. All information concerning the constitutive coefficients or the composite layering is input using this command.

Currently, this command supports only the solid elements type 17, type 23, type 24, type 25, type 72, type 91 and type 92, and the (Kirchhoff) shell elements type 20 and type 2020.

For the above solid elements, the constitutive matrix is inputted using the COEF sub-command. This matrix is defined in the local frame of the element specified in the CFRAMES command. Entries in the matrix that are not specified are determined from symmetry if appropriate, or are set to zero.

For the shell elements, the composite (or orthotropic) constitutive law can be prescribed either by entering the $C_{i,j}$ coefficients of the 6 by 6 constitutive matrix, or by defining the material properties and geometrical characteristics for each layer of the composite. In the first case, the sub-command COEF should be used. In the second case, either of the sub-commands LAYC or LAYN can be used, depending on whether the coupling between bending and membrane effects is to be enforced or not, respectively. When many layers of the composite are made of the same orthotropic material, the sub-commands LAYD and LAYMAT can be used instead of the sub-command LAYC to simplify the data entry process. Likewise, in similar circumstances, the sub-command LAYO and LAYMAT can be used instead of the sub-command LAYN.

The input format for the COMPOSITE command is given below.

COMPOSITE

COEF attribute#		
ROW# (i)	COLUMN# (j)	VALUE ($C_{i,j}$)
.		
.		
.		
ROW# (i)	COLUMN# (j)	VALUE ($C_{i,j}$)

COEF	Sub-command keyword used to directly input the coefficients of the constitutive law. Since the 6 by 6 constitutive matrix is symmetric, up to 21 independent coefficients may follow.
attribute#	Integer value that corresponds to the composite (or orthotropic or anisotropic) attribute of the element.
ROW#	Integer value that corresponds to the row i of the coefficient $C_{i,j}$. A row can be skipped if all its entries are zero.
COLUMN#	Integer value that corresponds to the column j of the coefficient $C_{i,j}$.
VALUE	Real value of the coefficient $C_{i,j}$.

For a solid element, the constitutive matrix *C* relates the stresses to the engineering strains in the element's local frame system {x ; y ; z} as follows:

$$\begin{pmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \tau_{xy} \\ \tau_{yz} \\ \tau_{xz} \end{pmatrix} = \begin{pmatrix} C_{1,1} & C_{1,2} & C_{1,3} & C_{1,4} & C_{1,5} & C_{1,6} \\ C_{1,2} & C_{2,2} & C_{2,3} & C_{2,4} & C_{2,5} & C_{2,6} \\ C_{1,3} & C_{2,3} & C_{3,3} & C_{3,4} & C_{3,5} & C_{3,6} \\ C_{1,4} & C_{2,4} & C_{3,4} & C_{4,4} & C_{4,5} & C_{4,6} \\ C_{1,5} & C_{2,5} & C_{3,5} & C_{4,5} & C_{5,5} & C_{5,6} \\ C_{1,6} & C_{2,6} & C_{3,6} & C_{4,6} & C_{5,6} & C_{6,6} \end{pmatrix} \begin{pmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_z \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{yz} \end{pmatrix}$$

For a (Kirchhoff) shell element, the constitutive matrix *C* relates the forces and moments to the mid-surface strains and curvatures in the shell element's local frame system {x ; y ; z} as follows:

$$\begin{pmatrix} N_x \\ N_y \\ N_{xy} \\ \mathcal{M}_x \\ \mathcal{M}_y \\ \mathcal{M}_{xy} \end{pmatrix} = \begin{pmatrix} C_{1,1} & C_{1,2} & C_{1,3} & C_{1,4} & C_{1,5} & C_{1,6} \\ C_{1,2} & C_{2,2} & C_{2,3} & C_{2,4} & C_{2,5} & C_{2,6} \\ C_{1,3} & C_{2,3} & C_{3,3} & C_{3,4} & C_{3,5} & C_{3,6} \\ C_{1,4} & C_{2,4} & C_{3,4} & C_{4,4} & C_{4,5} & C_{4,6} \\ C_{1,5} & C_{2,5} & C_{3,5} & C_{4,5} & C_{5,5} & C_{5,6} \\ C_{1,6} & C_{2,6} & C_{3,6} & C_{4,6} & C_{5,6} & C_{6,6} \end{pmatrix} \begin{pmatrix} \epsilon_x^0 \\ \epsilon_y^0 \\ \gamma_{xy}^0 \\ \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{pmatrix}$$

The element's local axes are defined with respect to the global reference frame by the three vectors defined in the CFRAMES command for the corresponding composite (orthotropic, or anisotropic) frame number that is specified in the ATTRIBUTES section. For shell elements, the constitutive matrix, *C*, can be decomposed into sub-matrices containing the bending and membrane properties of the shell element:

$$C = \begin{pmatrix} C_{mm} & C_{mb} \\ C_{bm} & C_{bb} \end{pmatrix}$$

Note the absence of the transverse shear behavior, as can be expected from a Kirchhoff type shell element. In the case of an isotropic material, or a single layer orthotropic or orthotropic material, there is no coupling between the bending and membrane behavior:

$$C_{mb} = C_{bm} = 0_{3\times 3}$$

For an isotropic material, the membrane constitutive matrix can be defined in terms of the Young's modulus, *E*, Poisson's ratio, ν,

and shell thickness, *h* as:

$$C_{mm} = \begin{pmatrix} \frac{Eh}{(1-\nu^2)} & \frac{\nu Eh}{(1-\nu^2)} & 0 \\ \frac{\nu Eh^3}{12(1-\nu^2)} & \frac{Eh^3}{12(1-\nu^2)} & 0 \\ 0 & 0 & \frac{Eh}{2(1+\nu)} \end{pmatrix}$$

while the bending constitutive matrix is given by:

$$C_{bb} = \begin{pmatrix} \frac{Eh^3}{12(1-\nu^2)} & \frac{\nu Eh^3}{12(1-\nu^2)} & 0 \\ \frac{\nu Eh}{(1-\nu^2)} & \frac{Eh}{(1-\nu^2)} & 0 \\ 0 & 0 & \frac{Eh^3}{24(1+\nu)} \end{pmatrix}$$

Using the COEF sub-command, different values of *E*, ν , and *h* can be utilized for building the membrane and bending components of the constitutive law, for example, if one wishes to adjust in some specific manner the bending and membrane behaviors of the shell element.

When the sub-command keyword COEF is used, the MATERIAL (see MATERIAL) command must be used as follows: the **density per unit surface** must be input at the fourth position after the attribute number, and for shells, the total thickness must also be defined at the seventh position after the attribute number if a stress analysis is requested.

For shell elements, if the pre-integrated constitutive matrix *C* is not available, the sub-command keywords LAYC or LAYN (or their counterparts LAYD, LAYO and LAYMAT) should be used to input the material properties and geometrical characteristics of each composite layer. If either LAYN or LAYO is used, no coupling between bending and membrane effects is enforced explicitly during integration through the thickness — that is: $C_{1,4} = C_{1,5} = C_{1,6} = 0$, $C_{2,4} = C_{2,5} = C_{2,6} = 0$ and $C_{3,4} = C_{3,5} = C_{3,6} = 0$.

The input format given below is the same for both sub-commands LAYC and LAYN.

LAYC (or LAYN)	attribute#								
k	$E_1^{(k)}$	$E_2^{(k)}$	$\nu_{12}^{(k)}$	$G_{12}^{(k)}$	$\mu_{1,12}^{(k)}$	$\mu_{2,12}^{(k)}$	$\rho^{(k)}$	$h^{(k)}$	$\theta_F^{(k)}$
.									
.									
.									
k	$E_1^{(k)}$	$E_2^{(k)}$	$\nu_{12}^{(k)}$	$G_{12}^{(k)}$	$\mu_{1,12}^{(k)}$	$\mu_{2,12}^{(k)}$	$\rho^{(k)}$	$h^{(k)}$	$\theta_F^{(k)}$

The input format given below is the same for both sub-commands LAYD and LAYO.

LAYD (or LAYO)	attribute#

k LAYER_MATERIAL_ID $h^{(k)}$ $\theta_F^{(k)}$

LAYMAT

LAYER_MATERIAL_ID $E_1^{(k)}$ $E_2^{(k)}$ $\nu_{12}^{(k)}$ $G_{12}^{(k)}$ $\mu_{1,12}^{(k)}$ $\mu_{2,12}^{(k)}$ $\rho^{(k)}$

or

LAYER_MATERIAL_ID $E_1^{(k)}$ $E_2^{(k)}$ $\nu_{12}^{(k)}$ $G_{12}^{(k)}$ $\rho^{(k)}$

.
.
.

LAYER_MATERIAL_ID $E_1^{(k)}$ $E_2^{(k)}$ $\nu_{12}^{(k)}$ $G_{12}^{(k)}$ $\mu_{1,12}^{(k)}$ $\mu_{2,12}^{(k)}$ $\rho^{(k)}$

or

LAYER_MATERIAL_ID $E_1^{(k)}$ $E_2^{(k)}$ $\nu_{12}^{(k)}$ $G_{12}^{(k)}$ $\rho^{(k)}$

LAYC, LAYD, LAYN, or LAYO — Sub-command keyword used for inputting geometrical and material properties for each layer of a composite shell element (characters).

attribute# — Integer value that corresponds to the composite (orthotropic) attribute of the element.

k — Integer value that corresponds to the layer number.

$E_1^{(k)}$ — Young's modulus in the local direction $1^{(k)}$ (that is, in the direction of the fibers for the k^{th} layer).

$E_2^{(k)}$ — Young's modulus in the local direction $2^{(k)}$ (that is, in the direction orthogonal to the fibers for the k^{th} layer).

$\nu_{12}^{(k)}$ — Poisson's ratio for transverse strain in the local direction $2^{(k)}$ when stressed in the local direction $1^{(k)}$ for $\sigma_1^{(k)}$ = constant and all other stresses zero.

$G_{12}^{(k)}$ — Transverse shear modulus in the plane $1^{(k)} - 2^{(k)}$ of the layer.

$\mu_{1,12}^{(k)}$ — Coefficient of mutual influence of the first kind which characterizes stretching in the local direction $1^{(k)}$ caused by shear in the plane $1^{(k)} - 2^{(k)}$ of the layer for $\tau_{12}^{(k)}$ = constant and all other stresses zero.

$\mu_{2,12}^{(k)}$ — Coefficient of mutual influence of the first kind which characterizes stretching in the local direction $2^{(k)}$ caused by shear in the plane $1^{(k)} - 2^{(k)}$ of the layer for $\tau_{12}^{(k)}$ = constant and all other stresses zero.

$\rho^{(k)}$ — Density (mass per unit volume) of the material of the k^{th} layer.

$h^{(k)}$ — Thickness of the k^{th} layer.

$\theta_F^{(k)}$ — Angle between a reference vector and the fibers in the layer that defines the orientation of these fibers. If CMP_FRM# is specified as an attribute for an element containing this layer under the command ATTRIBUTES, the reference vector is the projection onto the plane of that element of the first of the three vectors defining a local frame for this element and specified in the CFRAMES command. On the other hand, if a reference angle θ_{Ref}

is specified as an attribute for an element containing this layer under the command ATTRIBUTES, the reference vector is the vector obtained by rotating the directional edge connecting local nodes 1 and 2 of this element around its normal by an angle equal to θ_{Ref} . In both cases, this angle must be inputed in degrees.

LAYER_MATERIAL_ID — Identifier of a set of material properties (integer).

LAYMAT — Sub-command keyword that can be used for inputting the properties of a layer when it is made of a two-dimensional orthotropic material (characters).

The report number CU-CAS-94-16, “The 3-node Composite Shell and Isotropic Timoshenko Beam Elements” by Fran\c cois M. Hemez, provides a detailed description of the theory and implementation for the type-20 composite (orthotropic) shell element.

Next: CONDITION, Previous: COMPOSITE

23 COMPOSITE (ORTHOTROPIC SHELL OR ANISOTROPIC SOLID) ELEMENT FRAMES

Command Statement:	**CFRAMES**

The CFRAMES command statement is used to specify the orientation of the properties of composite laminates and orthotropic or anisotropic elements with respect to the global frame.

The input format is given below with the number of lines equal to the number of different composite (or orthotropic or anisotropic) element frames referenced in the ATTRIBUTES command. The syntax for the user defined composite element frame is:

CFRAMES

CMP_FRM#	S_{1x} S_{1y} S_{1z}	S_{2x} S_2 S_{2z}	S_{3x} S_{3y} S_{3z}

CMP_FRM# — Identification of a composite frame (integer).

S_{1x} S_{1y} S_{1z} — The first axis, **the local x-axis**, expressed in the global frame. For LAYC and LAYN type composites, the orientation of the fibers of the layers should be defined with respect to the projection of this axis onto the plane of the element.

S_{2x} S_{2y} S_{2z} — The second axis of the frame expressed in the global frame (floats).

S_{3x} S_{3y} S_{3z} — The third axis of the frame expressed in the global frame (floats).

Next: CONSTRAINTS, Previous: CFRAMES

24 CONDITION NUMBER

Command Statement:	**CONDITION**

The CONDITION command statement is used to request the evaluation of the condition number of the system being solved by FEM. The input format is given below.

CONDITION

TOLERANCE MAXITR

TOLERANCE	Error tolerance for computing the lowest and highest eigenvalues of the finite element model using the inverse power and power methods, respectively (float). The default value is 10^{-3} .
MAXITR	Maximum number of iterations for computing the lowest and highest eigenvalues of the finite element model using the inverse power and power methods, respectively (integer). The default value is 100 .

Next: CONTACTSURFACES, Previous: CONDITION

25 CONSTRAINTS

Command Statement:	**CONSTRAINTS**

The CONSTRAINTS command statement is used to specify a method for enforcing constraints defined in a problem. Its input format is given below.

Note 1: The direct elimination method for enforcing constraints is currently implemented for linear static and dynamic computations only.

Note 2: The Lagrange multiplier method for enforcing constraints is currently available for all but explicit dynamic analysis. It leads to an indefinite system of equations. Currently, only the spooles and mumps direct solvers with pivoting turned on and the gmres and FETI DP iterative solvers support this approach for solving constrained systems (see STATICS).

Note 3: The penalty method for enforcing constraints is currently available only for nonlinear static and dynamic analyses.

CONSTRAINTS

METHOD

METHOD	Specified default method for enforcing constraints (characters). This specified method can be overruled in the CONTACTSURFACES, MATERIAL, LMPC, NODALCONTACT, and TIEDSURFACES

commands.

`multipliers` In this case, which is also the default case, **FEM** uses the Lagrange multiplier method for enforcing all constraints associated with the LMPC, NODALCONTACT, CONTACTSURFACES, and TIEDSURFACES commands, and with joint and rigid elements (see TOPOLOGY).

`elimination` In this case, **FEM** uses an elimination method for enforcing all constraints associated with the LMPC and TIEDSURFACES commands, and with joint and rigid elements (see TOPOLOGY).

`penalty` [`beta`] In this case, **FEM** uses the penalty method for enforcing all constraints associated with the LMPC, NODALCONTACT, CONTACTSURFACES, and TIEDSURFACES commands, and with joint and rigid elements (see TOPOLOGY). The parameter `beta` should be a large positive number, typically of the order of 10^8 (no default value is provided).

Next: CONTROL, Previous: CONSTRAINTS

26 CONTACT SURFACES

Command Statement:	**CONTACTSURFACES**

The `CONTACTSURFACES` command statement can be used to enforce contact laws between pairs of surfaces defined using the SURFACETOPO command, for static and dynamic analyses only. Surface interactions are detected using the ACME library's search module.

`Note 1`: Setting the master and slave contact surfaces to the same surface activates the numerical treatement of self-contact.

`Note 2`: For the enforcement of contact surface constraints by the Lagrange multiplier method in explicit dynamic analysis, the discrete kinematic constraint equations are defined using a node-to-segment approach, and the contact forces are computed using the enforcement module of ACME. Four enforcement models are supported in this case and this case only: frictionless (default), constant friction, velocity dependent friction, and pressure dependent friction.

`Note 3`: For the enforcement of contact surface constraints by either the Lagrange multiplier method in static and implicit dynamic analyses, or the penalty method in static, explicit and implicit dynamic analyses, othe discrete kinematic constraint equations are defined using a segment-to-segment (mortar) approach. In these cases, only the frictionless contact model is supported.

`Note 4`: The enforcement of contact surface constraints by the Lagrange multiplier method in static and implicit dynamic analyses is supported only by the FETI-DP family of solvers.

`Note 5`: The enforcement of contact surface constraints by the elimination method is not supported.

```
CONTACTSURFACES
```

```
SURF_PAIR_ID#    MASTER    SLAVE    CONSTRAINT_METHOD
```

or, for statics or dynamics with an implicit scheme

```
SURF_PAIR_ID#    MASTER    SLAVE    CONSTRAINT_METHOD    MORTAR_TYPE    NORMAL_TOL    TANGENTIAL_TOL
```

or, for dynamics with an explicit scheme and the frictionless enforcement model

```
SURF_PAIR_ID#    MASTER    SLAVE    CONSTRAINT_METHOD    KPART_TYPE    NORMAL_TOL    TANGENTIAL_TOL    NUM_ITER
CONVERG_TOL
```

or, for dynamics with an explicit scheme and the constant friction enforcement model

```
SURF_PAIR_ID#    MASTER    SLAVE    CONSTRAINT_METHOD    KPART_TYPE    NORMAL_TOL    TANGENTIAL_TOL    NUM_ITER
CONVERG_TOL    FRIC_COEF
```

or, for dynamics with an explicit scheme and the velocity dependent friction enforcement model

```
SURF_PAIR_ID#    MASTER    SLAVE    CONSTRAINT_METHOD    KPART_TYPE    NORMAL_TOL    TANGENTIAL_TOL    NUM_ITER
CONVERG_TOL    STATIC_COEF    DYNAMIC_COEF    VELOCITY_DECAY
```

or, for dynamics with an explicit scheme and the pressure dependent friction enforcement model

```
SURF_PAIR_ID#    MASTER    SLAVE    CONSTRAINT_METHOD    KPART_TYPE    NORMAL_TOL    TANGENTIAL_TOL    NUM_ITER
CONVERG_TOL    FRIC_COEF    REF_PRES    OFFSET_PRES    PRES_EXP
```

SURF_PAIR_ID#		Id number of the surface pair to be described (integer).
MASTER		Identification of the master (mortar method) surface (see SURFACETOPO) (integer).
SLAVE		Identification of the slave (mortar method) surface (see SURFACETOPO) (integer).
CONSTRAINT_METHOD		Method for enforcing the associated constraints (characters). The default method is set in CONSTRAINTS and used whenever this entry is omitted.
	multipliers	The Lagrange multiplier method.
	elimination	The elimination method.
	penalty [beta]	The penalty method. The parameter beta should be a large positive number, typically of the order of 10^8 (no default value is provided).
MORTAR_TYPE		Mortar type: 0 = standard, 1 = dual, default value is 0 (integer).
NORMAL_TOL		Normal search tolerance used by ACME to identify interactions, default value is 0.1 (float) (see Figs. 1.2 and 1.3 in Section 1.3 of ACME's User Reference Manual).
TANGENTIAL_TOL		Tangential search tolerance used by ACME to identify interactions, default value is 0.001 (float) (see Figs. 1.2 and 1.3 in Section 1.3 of ACME's User Reference Manual).
KPART_TYPE		Kinematic partitioning type for explicit dynamics: 0 = fixed, 1 = automatic, default value is 0 (integer).
NUM_ITER		Maximum number of predictor-corrector iterations to be performed at each time step for explicit dynamics, default value is 5 (integer).
CONVERG_TOL		Convergence tolerance of the predictor-correctior iteration loop for explicit dynamics, default value is 1.0e-10 (float).
FRIC_COEF		Friction coefficient (float).
STATIC_COEF		Static friction coefficient (float).
DYANAMIC_COEF		Dynamic friction coefficient (float).

VELOCITY_DECAY	Velocity decay parameter (float).
REF_PRES	Reference pressure (float).
OFFSET_PRES	Offset pressure (float).
PRES_EXP	Pressure exponent (float).

Next: DECOMPOSE, Previous: CONTACTSURFACES

27 CONTROL STATEMENT

Command Statement:	**CONTROL**

FEM always generates a performance file named FNAME.timing which reports on the complexity, memory, and CPU resources associated with the performed computation. By default, the prefix FNAME is set to the name of the **FEM** input file (**FEM** also outputs some of these performance results on the screen). This optional command statement can be used to reset the value of the prefix FNAME. This prefix is also used by **FEM**, when executed with the "-t" (or "-T") option, to name the ASCII output file (FNAME.top) containing the geometry of the performed computation in the **XPost** format. It is also used by **FEM** for outputting, when requested, a number of domain decomposition (or mesh partitioning) files (see DECOMPOSE).

```
CONTROL
```

```
FNAME
ANATYPE
NODESET
ELEMSET
```

FNAME	Prefix of the names of the files reporting on the performance, geometry, and mesh decomposition of the performed computation, when applicable. The default value is the name of the **FEM** input file.
ANATYPE	Analysis type identifier (integer). Must be set to 2 for heat conduction analysis, and to 1 otherwise.
NODESET	Name of the node set describing the grid points of the geometry of the computation in the file FNAME.top outputted in the **XPost** format, when **FEM** is executed with the "-t" (or "-T") option. The default name is "nodes".
ELEMSET	Name of the element set describing the connectivity of the geometry of the computation in the file FNAME.top outputted in the **XPost** format, when **FEM** is executed with the "-t" (or "-T") option. The default name is "elems".

Next: DIMASS, Previous: CONTROL

28 DECOMPOSE

Command Statement:	**DECOMPOSE**

The DECOMPOSE command statement can be used to perform an element-based mesh partitioning (or domain decomposition) of the

computational domain, and exit or continue with a finite element analysis. The mesh partitioning algorithm is organized in two steps. The first step, which is based on the Greedy algorithm, generates an initial mesh partition. The second step optimizes the subdomain aspect ratios for the benefit of the FETI iterative solution algorithms.

Note 1: This command requires the presence in the FEM input file of a command specifying a FETI solver. When specified in the input file, this command is governed by the following format.

```
DECOMPOSE
```

```
NSUBS          nsubs
DETER
OUTFILE decomposition_filename
OUTLOAD
OUTMEM
EXIT
SKIP
```

NSUBS	
nsubs	Number of subdomains. Default is 1.
DETER	By default, the second step of the decomposition is carried out by a simulated annealing procedure. This sub-command keyword requests instead a deterministic approach to reduce the CPU time associated with the second step of the decomposition.
OUTFILE	
decomposition_filename	Requests outputing the decomposition in `decomposition_filename`. The format of this output is essentially the number and list of the elements in each subdomain. Default is `FNAME.optDec` where `FNAME` is the prefix specified under CONTROL (or its default value).
OUTLOAD	Requests outputing the load distribution in file `decomposition_filename.load`. The load distribution is based on the weights of the elements in the subdomains (see WEIGHTS). Default is off.
OUTMEM	Requests outputing in file `decomposition_filename.mem` an estimate of the subdomain-based memory comsumption. This estimate is based on the requirements for factoring the subdomain matrices by a direct skyline method. By default, mid-side nodes are ignored by the overall mesh partitioning algorithm. However, when this option is specified, mid-side nodes are taken into account. Default is off.
EXIT	Requests exiting after the mesh partition is generated. Default is off.
SKIP	Requests skipping this command. Default is off.

When specified on the command line, the syntax of this command is as follows.

`fem_executable --dec --nsub` *[number_of_subdomains]* `--deter --load --mem --exit` <*FEM_input_filename*> where `--dec` requests a mesh decomposition and `--nsub` specifies the number of subdomains. In this case, the prefix for the names of the files associated with the mesh decomposition and analysis is the content of the `FNAME` entry of the CONTROL command or its default value.

Note 1: When this command is executed from the command line and specified in the input file, any parameter specified on the command line overrides its value specified in the input file.

Note 2: When a decomposition file is already available, the syntax for running FEM with this decomposition file on a shared memory is

`fem_executable -d` *[decomposition_filename]* `-v` *[verbose_frequency]* `-n` *[decomposition_filename]* <*FEM_input_filename*>

The above command can be combined with the decomposition command to obtain

`fem_executable --dec --nsub` *[number_of_subdomains]* `--load --mem` *[decomposition_filename]* `-v` *[verbose_frequency]* `-n` *[decomposition_filename]* <*FEM_input_filename*>

Next: DYNAMICS, Previous: DECOMPOSE

29 DISCRETE NODAL MASS AND INERTIA

Command Statement:	**DIMASS**

The `DIMASS` command statement is used to lump a discrete mass or inertia on a specified degree of freedom of a specified node. If a node number appears more than once under this command with the same degree of freedom, FEM sums all the lumped masses at the specified degree of freedom of the specified node. If a gravity field is also specified in the input file using the `GRAVITY` command, FEM generates at each specified degree of freedom of each specified node under this command a weight-force component equal to the product of the specified discrete mass and specified gravity acceleration in the corresponding direction.

The input format is given below.

```
DIMASS
```

```
NODE#        DOF#        VALUE
```

`NODE#`	Node number where the mass or inertia will be added (integer).
`DOF#`	Degree of freedom where the mass or inertia will be added (integer).
`VALUE`	Value of the prescribed discrete mass or inertia (float).

If more than one line addressing the same degree of freedom of the same node are encountered under this command, the effect is the accumulation of the lumped masses or inertias at this degree of freedom.

Next: EIGEN, Previous: DIMASS

30 DYNAMIC ANALYSIS

Command Statement:	**DYNAMICS**

The DYNAMICS command is used to specify a time-integrator for a desired dynamic (transient, time-domain) structural, acoustic, or heat transfer (thermal) analysis and other relevant parameters. Currently, the following time-integrators are available in **FEM**:

1. The implicit generalized α method and the explicit central difference method for linear and nonlinear (in this case, the NONLINEAR command must also be present in the input file) structural dynamic analyses.
2. The implicit generalized α method, the explicit central difference method, and the explicit modified wave equation algorithm for linear, time-domain acoustic analysis.
3. The generalized midpoint family of methods for linear heat transfer analysis.
4. The implicit midpoint rule for nonlinear heat transfer analysis.

Linear and Nonlinear Structural Dynamic Analyses and Linear Time-Domain Acoustic Analyses

- The implicit generalized α method for linear and nonlinear structural dynamic and linear time-domain acoustic analyses enforces the following equation of equilibrium

$$M\ddot{u}^{n+1-\alpha_m} + C\dot{u}^{n+1-\alpha_f} + F_{internal}(u^{n+1-\alpha_f}) = F_{external}^{n+1-\alpha_f}$$

where

$$u^{n+1-\alpha_f} = (1-\alpha_f)u^{n+1} + \alpha_f u^n$$

$$\dot{u}^{n+1-\alpha_f} = (1-\alpha_f)\dot{u}^{n+1} + \alpha_f \dot{u}^n$$

$$\ddot{u}^{n+1-\alpha_m} = (1-\alpha_m)\ddot{u}^{n+1} + \alpha_m \ddot{u}^n$$

$$F_{external}^{n+1-\alpha_f} = F_{external}(t^{n+1-\alpha_f})$$

$$t^{n+1-\alpha_f} = (1-\alpha_f)t^{n+1} + \alpha_f t^n$$

and for linear problems

$$F_{internal}(u^{n+1-\alpha_f}) = Ku^{n+1-\alpha_f}$$

Hence, the above equation of equilibrium can also be written as

$$M\left((1-\alpha_m)\ddot{u}^{n+1} + \alpha_m\ddot{u}^n\right) + C\left((1-\alpha_f)\dot{u}^{n+1} + \alpha_f\dot{u}^n\right) + F_{internal}\left((1-\alpha_f)u^{n+1} + \alpha_f u^n\right) = F_{external}^{n+1-\alpha_f}$$

Given $F_{external}(t)$, u^0 and $\dot{u}^0$, the implicit generalized α method solves the above equation as follows

$$\ddot{u}^0 = M^{-1}(F_{external}^0 - F_{internal}(u^0) - C\dot{u}^0)$$

$$u^{n+1} = u^n + \Delta t\dot{u}^n + \Delta t^2\left(\left(\tfrac{1}{2}-\beta\right)\ddot{u}^n + \beta\ddot{u}^{n+1}\right)$$

$$\dot{u}^{n+1} = \dot{u}^n + \Delta t\left((1-\gamma)\ddot{u}^n + \gamma\ddot{u}^{n+1}\right)$$

where $\beta \neq 0$. The Newmark method is obtained by setting $\alpha_m = \alpha_f = 0$. The midpoint rule is obtained by setting $\alpha_m = \alpha_f = \frac{1}{2}$, $\beta = \frac{1}{4}$ and $\gamma = \frac{1}{2}$. The *classical* central difference method is obtained by setting $\alpha_m = \alpha_f = 0$, $\beta = 0$ and $\gamma = \frac{1}{2}$.

The Newmark method ($\alpha_m = \alpha_f = 0$) is second-order time-accurate if and only if $\gamma = \frac{1}{2}$. For undamped linear systems, it is unconditionally stable if and only if $\gamma \geq \frac{1}{2}$ and $\beta \geq \frac{1}{4}(\gamma + \frac{1}{2})^2$.

The generalized α method is second-order time-accurate if and only if $\gamma = \frac{1}{2} - \alpha_m + \alpha_f$. For undamped systems, it is unconditionally stable for linear problems if and only if $\alpha_m \leq \alpha_f \leq \frac{1}{2}$ and $\beta \geq \frac{1}{4} + \frac{1}{2}(\alpha_f - \alpha_m)$. It maximizes high-frequency dissipation when

$$\beta = \tfrac{1}{4}(1 - \alpha_m + \alpha_f)^2, \quad \alpha_f = \tfrac{\rho_\infty}{\rho_\infty + 1}, \quad \alpha_m = \tfrac{2\rho_\infty - 1}{\rho_\infty + 1}$$

where ρ_∞ is the user-specified high-frequency dissipation ($0 \leq \rho_\infty \leq 1$).

- The central difference method for linear and nonlinear structural dynamic and time-domain acoustic analyses implemented in **FEM** enforces the following equation of equilibrium

$$M\ddot{u}^{n+1} + C\dot{u}^{n+\frac{1}{2}} + F_{internal}(u^{n+1}) = F_{external}^{n+1}$$

where

$$\dot{u}^{n+\frac{1}{2}} = \dot{u}^n + \tfrac{\Delta t}{2}\ddot{u}^n$$

Given $F_{external}(t)$, u^0 and $\dot{u}^0$, it solves the above equation as follows

$$\ddot{u}^0 = M^{-1}(F_{external}^0 - F_{internal}(u^0) - C\dot{u}^0)$$

$$u^{n+1} = u^n + \Delta t\dot{u}^n + \tfrac{\Delta t^2}{2}\ddot{u}^n$$

$$\dot{u}^{n+1} = \dot{u}^n + \tfrac{\Delta t}{2}(\ddot{u}^n + \ddot{u}^{n+1})$$

This central difference method for linear and nonlinear structural dynamic and linear time-domain acoustic analyses computes the acceleration as follows

$$\ddot{u}^{n+1} = M^{-1}\left(F_{external}^{n+1} - F_{internal}(u^{n+1}) - C(\dot{u}^n + \tfrac{\Delta t}{2}\ddot{u}^n)\right)$$

For this reason, **FEM** automatically lumps the mass matrix M when this method is selected so that it becomes explicit.

- The explicit modified wave equation method for linear time-domain acoustic analysis implemented in **FEM** enforces the following equation of equilibrium

$$M\ddot{u}^{n+1} + C\dot{u}^{n+1} + (K - \tfrac{\Delta t^2}{q} K M^{-1} K)u^{n+1} = F^{n+1}$$

where C is the damping matrix associated with a non-absorbing boundary condition when such a condition is imposed, and q is a real-valued parameter.

Given $F(t)$, u^0 and $\dot{u}^0$, it solves the above equation as follows

$$\ddot{u}^0 = M^{-1}\left(F^0 - C\dot{u}^0 - K(u^0 - \tfrac{\Delta t^2}{q} M^{-1} K u^0)\right)$$

$$\dot{u}^{\frac{1}{2}} = \dot{u}^0 + \tfrac{\Delta t}{2}\ddot{u}^0$$

$$u^{n+1} = u^n + \Delta t \dot{u}^{n+\frac{1}{2}}$$

$$G^{n+1} = K\left(I - \tfrac{\Delta t^2}{q}(M + \tfrac{\Delta t}{2}C)^{-1}K\right)u^{n+1}$$

$$\dot{u}^{n+\frac{3}{2}} = (M + \tfrac{\Delta t}{2}C)^{-1}\left(M\dot{u}^{n+\frac{1}{2}} + \tfrac{\Delta t}{2}(2F^{n+1} - 2G^{n+1} - C\dot{u}^{n+\frac{1}{2}})\right)$$

For $C = 0$ and $q = 12$, the explicit modified wave equation method is fourth-order time-accurate. For $C = 0$ and $12 \leq q \leq 12.5$, experience reveals that for many applications, this method delivers an almost sixth-order time-accuracy. For $C \neq 0$, the explicit modified wave equation method is second-order time-accurate.

Linear Heat Transfer Analysis

- The generalized midpoint family of methods for linear heat transfer analysis enforces the equation of equilibrium

$$Q\dot{u}^{n+\frac{1}{2}} + Hu^{n+\frac{1}{2}} = F^{n+\frac{1}{2}}$$

where

$$u^{n+\frac{1}{2}} = \tfrac{1}{2}(u^{n+1} + u^n)$$

$$\dot{u}^{n+\frac{1}{2}} = \tfrac{1}{2}(\dot{u}^{n+1} + \dot{u}^n)$$

and solves it as follows

$$\dot{u}^0 = Q^{-1}(F^0 - Hu^0)$$

$$\dot{u}^{n+\alpha} = \alpha\dot{u}^{n+1} + (1-\alpha)\dot{u}^n$$

$$u^{n+1} = u^n + \Delta t\left(\alpha\dot{u}^{n+1} + (1-\alpha)\dot{u}^n\right)$$

The generalized midpoint family of methods outlined above is unconditionally unstable for $\alpha > \frac{1}{2}$. Hence, only the choices satistying $\alpha \leq \frac{1}{2}$ are allowed. The forward Euler method is obtained for $\alpha = 0$. The backward Euler implicit method is obtained for $\alpha = 1$. The implicit trapezoidal rule is obtained for $\alpha = \frac{1}{2}$.

Nonlinear Heat Transfer Analysis

- The implicit midpoint rule for nonlinear heat transfer analysis enforces the following equation of equilibrium

$$Q\dot{u}^{n+\frac{1}{2}} + F_{internal}\left(u^{n+\frac{1}{2}}\right) = F_{external}^{n+\frac{1}{2}}$$

where

$$u^{n+\frac{1}{2}} = \frac{1}{2}\left(u^{n+1} + u^{n}\right)$$

$$\dot{u}^{n+\frac{1}{2}} = \frac{1}{2}\left(\dot{u}^{n+1} + \dot{u}^{n}\right)$$

Given $F_{external}(t)$ and u^0 , it solves the above equation as follows

$$\dot{u}^0 = Q^{-1}\left(F_{external}^0 - F_{internal}(u^0)\right)$$

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

Note 1: Explicit time-integration algorithms are currently not supported for finite element models with rigid elements, massless (or moment-of-interia-less) degrees of freedom, or linear multipoint constraints.

Note 2: Whenever an implicit time-integration algorithm is selected, an equation solver must be specified under the STATICS command.

Note 3: For nonlinear heat transfer analysis (see NONLINEAR), the appropriate midpoint rule algorithm is automatically selected by this command.

The input format of this command is given below.

```
DYNAMICS
```

IACC	flagIACC				
STABLE	flagSTABLE	TOLERANCE	MAXITR		
MODAL	(optional)				
MECH	β	γ			(midpoint implementation of Newmark — that is, $\alpha_m = \alpha_f = \frac{1}{2}$)
MECH	β	γ	α_f	α_m	(generalized Ω)
MECH	ρ_∞				(generalized Ω)
ACOU	β	γ			(midpoint implementation of Newmark — that is, $\alpha_m = \alpha_f = \frac{1}{2}$)
ACOU	β	γ	α_f	α_m	(generalized Ω)
ACOU	ρ_∞				(generalized Ω)
ACOU	β	γ	q		(modified wave equation method)
HEAT	Ω				
TIME	TH	TM	TT		
DAMP	*a*	*b*			
DAMP	MODAL				
MODE#	MDV				
.					
.					
MODE#	MDV				

IACC — For linear and nonlinear explicit dynamic computations, and for nonlinear implicit dynamic computations, the initial acceleration is always computed to satisfy equilibrium at $t = 0$. For linear implicit dynamic computations, the initial acceleration is always set to zero if a FETI solver is specified under the STATICS command; otherwise, the user can request to have it either set to zero or computed to satisfy equilibrium at $t = 0$. IACC is the sub-command keyword that enables the user to make this choice in the latter case (characters).

flagIACC — On/Off flag (characters). The default value is on.

on — In this case, for linear implicit dynamic computations, as long as a FETI method is not chosen as the equation solver under the STATICS command, the acceleration is initialized to satisfy equilibrium at $t = 0$. This initialization requires the factorization of the mass matrix. The sparse direct solver is always used for this purpose, no matter which direct equation solver is specified under the STATICS command. If for some reason the finite element model has one or several massless or moment-of-inertia-less degrees of freedom, the sparse solver will typically encounter zero pivots and set the initial acceleration to zero at the corresponding degrees of freedom.

off — In this case, for linear implicit dynamic computations, the initial acceleration is set to zero.

STABLE — Sub-command keyword for managing the time-step of a linear or nonlinear, structural dynamic or acoustic time-integration using the central difference method (characters).

`flagSTABLE` On/Off flag (characters).

`on` In this case, which is also the default case, **FEM** computes automatically the critical time-step — that is, the maximum stability time-step — for the linear and nonlinear explicit central difference methods. For this purpose, **FEM** computes the largest eigenvalue of the finite element model using the power method. If the specified time-step TM (see below) is larger than the critical time-step, then the critical time-step is used instead for the computation.

`off` In this case, no check is performed for the maximum stability time-step and the time-step TM (see below) is used as specified during the computation.

`TOLERANCE` Error tolerance for computing the highest eigenvalue of the finite element model using the power method (float). The default value is 10^{-3}.

`MAXITR` Maximum number of iterations for computing the highest eigenvalue of the finite element model using the power method (integer). The default value is 100.

`MODAL` Sub-command keyword that requests a modal superposition dynamic analysis (characters). In this case, the READMODE command must be present in the input file to provide the eigenmodes to be used (characters). As in the case of a direct time-integration rather than modal analysis, the initial displacement (see IDISPLACEMENTS) and velocity (see IVELOCITIES) conditions can be input in either the global or generalized coordinates (or reduced system), or in both.

`MECH` or `ACOU` or `HEAT` Sub-command keyword `MECH` signals a structural dynamic analysis (characters). Sub-command keyword `ACOU` signals a time-domain acoustic analysis. Sub-command keyword `HEAT` signals a thermal analysis.

β Newmark coefficient β for a second-order system (real). Set $\beta = 0$ ($\alpha_m = \alpha_f = 0$) and $\gamma = \frac{1}{2}$ to specify the explicit central difference method.

γ Newmark coefficient γ for a second-order system (real).

α_f Generalized α coefficient α_f for a second-order system (real). To obtain the standard Newmark algorithm, set $\alpha_f = 0$. The default is $\alpha_f = \frac{1}{2}$.

α_m Generalized α coefficient α_m for a second-order system (real). To obtain the standard Newmark algorithm, set $\alpha_m = 0$. The default is $\alpha_m = \frac{1}{2}$.

ρ_∞ Generalized α infinite frequency spectral radius for a second-order system (real). When this option is used and this coefficient is specified, α_f, α_m, β and γ are automatically set to

$$\alpha_f = \frac{\rho_\infty}{\rho_\infty+1}, \quad \alpha_m = \frac{2\rho_\infty-1}{\rho_\infty+1}, \quad \beta = \frac{1}{4}(1-\alpha_m+\alpha_f)^2, \quad \gamma = \frac{1}{2}-\alpha_m+\alpha_f$$

, which results in a second-order unconditionally stable scheme that minimizes low frequency dissipation.

q For $q = 12$, the explicit modified wave equation method is fourth-order time-accurate. Furthermore, for many applications, experience shows that for $12 \leq q \leq 12.5$, this method behaves almost like a sixth-order time-accurate scheme.

`HEAT` Sub-command keyword that preceeds the parameter of the Newmark algorithm for a first-order time-dependent system (characters).

α Main parameter of the generalized trapezoidal family of methods for first-order systems (real). This parameter must be chosen within $\alpha \leq \frac{1}{2}$.

TIME	Sub-command keyword to specify time-interval parameters (characters).
TH	Integration time-step for heat transfer analysis (real). This time-step should be put to zero if a structural dynamic or time-domain acoustic analysis is to be performed.
TM	Integeration time-step for a structural dynamic or time-domain acoustic analysis (real).
TT	End of time-interval for time-integration (real); beginning of time-interval for time-integration is zero for all types of time-dependent analyses.
DAMP	Sub-command keyword to specify a Rayleigh damping (characters). This option is active only when MECH is specified.
a	Rayleigh damping stiffness coefficient a (real). This option is active only when MECH is specified. In the nonlinear case, this coefficient is assigned at each iteration to the linearized stiffness matrix and the product of these two quantities, which varies with the nonlinear iterations, constitutes the contribution to the damping term.
b	Rayleigh damping mass coefficient b (real). This option is active only when MECH is specified.
DAMP MODAL	Sub-command keywords to signal that the following lines specify modal damping ratios (characters). This option is active only when the MECH and MODAL sub-commands are also specified — that is, for a dynamic analysis of a mechanical system using mode superposition. If a mode is repeated, the modal damping ratio values are added for that mode. If both DAMP and DAMP MODAL commands are specified, and the keyword MODAL is also specified, then modal damping takes precedence over Rayleigh damping.
MODE#	Mode id number (integer).
MDV	Modal damping value (ξ) for mode MODE# (real).

Next: END, Previous: DYNAMICS

31 EIGENVALUE PROBLEMS

Command Statement:	**EIGEN**

The EIGEN command statement is used to request an eigenvalue analysis and specify its parameters. Currently, only the subspace iteration method (the default choice) and the ARPACK package are implemented and hooked in **FEM**.

The input format of this command is given below. An example input file can be found in FEM.d/fem_examples/Eigen.d.

Note 1: In principle, the usage of the EIGEN command requires the simultaneous usage of the STATICS command to specify an equation solver. By default, EIGEN selects the default of the STATICS command as an equation solver.

EIGEN

VERSION	
SHIFT	σ
NSBSPV	nsbspv
NEIGPA	neigpa

TOLEIG	toleig		
TOLJAC	toljac		
ARPACK	which		
ARPACK	lbound	nshifts	
ARPACK	lbound	ubound	neigps
MAXITR	maxitr		

VERSION

explicit — In this case, the "stiffness" matrix is used explicity when constructing the reduced eigen problem in the subspace iteration method. Otherwise (default), the reduced eigen problem is constructed by exploiting some mathematical identities to avoid using explicitly the "stiffness" matrix which has been factored by then. Note that specifying the explicit version of the subspace iteration algorithm incurs additional memory storage.

SHIFT

σ — Value of a specified shift for the "stiffness" matrix (real).

NSBSPV

nsbspv — In the case of the subspace iteration method (default eigensolver), nsbspv denotes the number of subspace iteration vectors and should be set to the minimum between twice the number of requested eigen pairs, $2\times$ neigpa, and neigpa + 8. If rigid body modes are expected, the previous formula should be increased by the total number of rigid body modes. In the case of ARPACK, nsbspv denotes the number of Krylov vectors (per shift, if a shifting strategy is used).

NEIGPA

neigpa — Number of requested eigen pairs. If rigid body modes are found, their number is included in the specified value of neigpa.

TOLEIG

toleig — Tolerance for the convergence of the subspace iteration method.

TOLJAC

toljac — Tolerance for the Jacobi Determinant algorithm used in the subspace iteration method.

ARPACK — The presence of this keyword under the EIGEN command specifies the usage of the ARPACK eigensolver in the shift-invert mode. Otherwise, the default choice is the subspace iteration method. The ARPACK choice is required when one or both of the two matrices governing the eigenvalue problem is/are indefinite. This is the case, for example, when the problem contains linear multipoint constraints, mortar elements, rigid elements, etc. In this case, it is also recommended to use the SPOOLES or MUMPS sparse direct solver.

which — Keyword (characters) to specify which eigenpairs to be computed by ARPACK using the shift-invert solution method. This keyword can take one of the following values.

LA — In this case, the neigpa eigenpairs whose eigenvalues are just to the right of the shift σ are computed. This is the default value of which when the shift is zero ($\sigma = 0$).

SA — In this case, the neigpa eigenpairs whose eigenvalues are just to the left of the shift σ are computed.

BE — In this case, neigpa eigenpairs with eigenvalues on either side of the shift σ are computed. This is the default value of which when the shift is non zero ($\sigma \neq 0$).

lbound — Lower bound of a set or range of eigenvalues to be computed by ARPACK (real).

nshifts	Number of shifts to be used by ARPACK when computing the neigpa eigenvalues that are greater than lbound (integer). The specific values of the shifts are automatically selected by ARPACK, and neigpa/nshifts eigenpairs are computed per shift.
ubound	Upper bound of a range of eigenvalues to be computed by ARPACK (real).
neigps	When a range of eigenvalues [lbound, ubound] is specified, ARPACK computes all eigenpairs whose eigenvalues lie within this range. In this case, a first shift is set to lbound and neigps consecutive eigenvalues within the range [lbound, ubound] are computed. Then, a recursive procedure in which the largest previously computed eigenvalue is chosen as a new shift and neigps new eigenvalues within the range [lbound, ubound] are computed using this shift is applied until all eigenvalues within the specified range are captured. The default value for neigps is 50.
MAXITR	
maxitr	Maximum number of iterations for the eigensolver. The default value is 10× nsbspv.

Next: FSINTERFACE, Previous: EIGEN

32 END

The END statement is used to indicate the end-of-file. It should always be the last statement.

Next: HFTT, Previous: END

33 FLUID/STRUCTURE INTERFACE

Command Statement:	**FSINTERFACE**

The FSINTERFACE command statement is used to specify a fluid/structure interface between two surfaces defined using the SURFACETOPO command (see SURFACETOPO). In this case, the treatment of this interface relies on the capabilities of the ACME library.

The input format of this command is given below.

```
FSINTERFACE
```

```
SURF_PAIR_ID#        FLUID_SURF        STR_SURF
```

or

```
SURF_PAIR_ID#        FLUID_SURF        STR_SURF        NORMAL_TOL        TANGENTIAL_TOL
```

SURF_PAIR_ID#	Id number of the surface pair to be described (integer).
FLUID_SURF	Identification of the fluid surface (see SURFACETOPO) (integer).
STR_SURF	Identification of the structure surface (see SURFACETOPO) (integer).

NORMAL_TOL	Normal search tolerance used by ACME to identify interactions, default value is 0.1 (float) (see Figs. 1.2 and 1.3 in Section 1.3 of ACME's User Reference Manual).
TANGENTIAL_TOL	Tangential search tolerance used by ACME to identify interactions, default value is 0.001 (float) (see Figs. 1.2 and 1.3 in Section 1.3 of ACME's User Reference Manual).

Next: MFTT, Previous: FSINTERFACE

34 FORCE TIME TABLE-HEAT CONDUCTION

Command Statement:	**HFTT**

The HFTT command statement must be used to specify the time-dependent amplification of the boundary fluxes. Skipping this statement for dynamics problems results in specifying unit boundary flux amplitudes. For FEM, pairs of time and amplification values are input. Linear interpolation is also used for “in between” points.

```
HFTT
```

```
TIME_1    AMP_1
.         .
.         .
.         .
TIME_n    AMP_n
```

TIME_1	A specified time point (float).
AMP_1	A specified amplification value at time point TIME_1 (float). This amplification factor is automatically set to zero for all times prior to the earliest specified time point and all times later than the latest specified time point.

Next: GRBM, Previous: HFTT

35 FORCE TIME TABLE-MECHANICS AND ACOUSTICS

Command Statement:	**MFTT**

The MFTT command statement can be used to implement time-dependent tensor-product forms of the boundary conditions enforced by the commands FORCE, ATDDNB, ATDROB, and PRESSURE. In the absence of this statement, these commands imply unit amplitudes. Pairs of time and amplification values are input. Linear interpolation is also used for “in between” points.

When applied with the FORCE command, the result is the amplification of the force value. When applied with the ATDDNB command, the result is the amplification of the resulting distributed Neumann boundary condition. When applied with the ATDROB command, the result is the amplification of the right hand-side of the distributed Robin boundary condition. A FEM example input file using this

command can be found in FEM.d/fem_examples/MFTT.d

```
MFTT
```

```
TIME_1     AMP_1
.          .
.          .
.          .
TIME_n     AMP_n
```

TIME_1	A specified time point (float).
AMP_1	A specified amplification value at time point TIME_1 (float). This amplification factor is automatically set to zero for all times prior to the earliest specified time point and all times later than the latest specified time point.

Next: GEPS, Previous: MFTT

36 GEOMETRIC RIGID BODY MODES

Command Statement:	**GRBM**

The GRBM command is useful for a number of analyses of unrestrained or partially restrained finite element models. It has two separate effects.

- 1. The GRBM command requests the computation of the rigid body modes (or zero energy modes in general) of a given (unrestrained, partially restrained, or even restrained) finite element model using the hybrid geometric-algebraic method published in “C. Farhat and M. Geradin, *On the General Solution by a Direct Method of a Large-Scale Singular System of Linear Equations: Application to the Analysis of Floating Structures*, International Journal for Numerical Methods in Engineering, Vol. 41, pp. 675-696 (1998). This hybrid method combines a geometry-based algorithm and the SVD factorization of a matrix of constraints corresponding to the given boundary conditions. The SVD factorization relies on the first tolerance specified under this command for identifying the deemed zero singular values, which are associated with the sought-after singular modes. Whenever **FEM** finds the GRBM in its input file, it computes the rigid body modes (or zero energy modes in general) of the given finite element model and prints on the screen the found number of rigid body or zero energy modes. The computed rigid body or zero energy modes themselves are used, for example, by the RBMFILTER command. They are also automatically included in the output of an eigen computation (see OUTPUT).

- 2. When the requested analysis incurs the solution of a singular system of finite element equations, the GRBM command requests the computation of the generalized inverse of the representative singular finite element matrix in factored form using the algorithm published also in “C. Farhat and M. Geradin, *On the General Solution by a Direct Method of a Large-Scale Singular System of Linear Equations: Application to the Analysis of Floating Structures*, International Journal for Numerical Methods in Engineering, Vol. 41, pp. 675-696 (1998). This algorithm relies on the information computed by the hybrid

geometric-algebraic method outlined above (number of rigid body or zero energy modes of the finite element model, and in some cases, the rigid body or zero energy modes themselves). Singular finite element systems arise in general in static problems, whether they are linear or nonlinear. However, tangent finite element matrices associated with unrestrained or partially restrained nonlinear finite element models can have a number of rigid body or zero energy modes that is different from that which can be deduced from the analysis of the matrix of constraints associated with the specified boundary conditions. For this reason, this second part of the GRBM command which selects the aforementioned algorithm for solving a singular system of equations takes effect only for **linear** static, quasistatic, eigen (structural), aeroelastic, and thermoaeroelastic analyses. When the STATICS command is specified in the input file, then: (a) if a direct solver is specified under this command, the generalized inverse of the stiffness matrix is computed in factored form using the aforementioned algorithm associated with the first part of the GRBM command, and its pivots are monitored for small values (deemed to be zero) using the second tolerance specified under the GRBM command — such deemed zero pivots are associated with zero energy modes associated with mechanisms or model oddities that cannot be detected by the hybrid geometric-algebraic method outlined above; (b) if a FETI-1 or FETI-2 solver is specified under the STATICS command, the rigid body modes of the floating subdomains are computed using the same hybrid geometric-algebraic method and their singular matrices are factored using the same aforementioned algorithm (see section on FETI parameters in STATICS); (c) if a FETI-DP solver is specified under the STATICS command, the tolerance for detecting and computing the rigid body modes of the global structure is set to the first tolerance specified under the GRBM command.

Note 1: See TRBM for an alternative option for analyzing singular systems.

Note 2: If both the GRBM and TRBM commands are specified in the same **FEM** input file, the one specified last is chosen. However, if that one is TRBM, it does not de-activate the GRBM command but simply overwrites the value of its second tolerance, VALUE_2, by the value of the tolerance specified under TRBM.

GRBM

VALUE_1 VALUE_2

VALUE_1 Tolerance for identifying the zero singular values of the constraint matrix analyzed by the hybrid geometric-algebraic method for finding the rigid body or zero energy modes of a finite element model (real). Its default value is 1.0e-6.

VALUE_2 Tolerance for capturing algebraically the mechanisms and other oddities that cause a finite element matrix to be singular and which are not visible by the geometric-algebraic method for finding the rigid body or zero energy modes of a finite element model (real). Its default value is 1.0e-16.

Next: GRAVITY, Previous: GRBM

37 GEOMETRIC STIFFENING DUE TO PRESTRESS

Command Statement: **GEPS**

The GEPS command statement is essentially a switch that turns on the accounting of pre-stress effects in the form of a geometric

stiffness matrix K_G . This geometric stiffness matrix is computed around a displacement field specified under the IDISP6 command which must be present in the input file. A sample input file illustrating the use of GEPS can be found in FEM.d/fem_examples /Buckle.d/

The effect of this switch on the EIGEN command statement is to replace the stiffness matrix by the tangent stiffness matrix and therefore compute the modes of the structure around the deformed position implied by the displacement field specified under the IDISP6 command.

Its effect on the DYNAMICS and STATICS commands is to replace the stiffness matrix by the tangent stiffness matrix around the displacement field specified under the IDISP6 command, including in aeroelastic applications.

Note 1: The GEPS command can be used only in a linear analysis.

Note 2: If an aeroelastic analysis is requested with the GEPS and IDISP6 commands present in the input file, and the IDISPLACEMENTS command is specified to initialize the displacement field, then FEM sends to the fluid code at each time step the sum of the updated displacement field (interpreted in that case as an increment displacement) and the displacement specified under the IDISP6 command. If in such a case the IDISPLACEMENTS command is not present in the input file, the displacement field of the structure is then initialized by the IDISP6 command, and FEM acts as follows: at the first time-step, it sends to the fluid code only the updated displacement field (since initialization accounts in that case for the content of the IDISP6 command), but at each subsequent time-step, it sends the sum of the updated displacement field and the displacement specified under the IDISP6 command.

GEPS

Next: GROUPS, Previous: GEPS

38 GRAVITATIONAL ACCELERATION

Command Statement:	**GRAVITY**

The GRAVITY command is used to specify directional gravitational acceleration constants. The input format is given below.

Note 1: By default, **FEM** computes all element gravity loads by a consistent approach, **except for explicit dynamic computations** (see DYNAMICS), in which case **FEM** always uses a lumped approach for this purpose. If a consistent mass matrix is not available for a particular element (see TOPOLOGY), then **FEM** uses in all cases a lumped gravity load for that element. Alternatively, the LUMPED command can be used to instruct **FEM** to compute all element mass matrices and gravity loads by a lumped approach.

GRAVITY

```
COEFF_X        COEFF_Y        COEFF_Z
```

COEFF_X	Gravitational Acceleration Constant in the X direction.
COEFF_Y	Gravitational Acceleration Constant in the Y direction.
COEFF_Z	Gravitational Acceleration Constant in the Z direction.

Next: HZEM, Previous: GRAVITY

39 GROUPS

Command Statement: **GROUPS**

The GROUPS command can be used to organize elements into groups according to their attribute number (see ATTRIBUTES), and nodes into groups according to their integer node identifying number. For example, the notion of a group of elements is used by the RANDOM command to attribute a defined random material property to a group of elements. That of a group of nodes is used by the OUTPUT command to output computed results for groups of nodes.

Note 1: Group identifiers must be contiguous integers starting from 1.

```
GROUPS
```

```
A        ATTRIBUTE#        GROUP#
```

or

```
A        STARTING_ATTRIBUTE#        ENDING_ATTRIBUTE#        GROUP#
```

or

```
N        NODE#        GROUP#
```

or

```
N        STARTING_NODE#        ENDING_NODE#        GROUP#
```

A	Sub-command keyword (character) signaling that elements are to be grouped according to their attribute number — that is, all elements whose attribute is the same as ATTRIBUTE# or within the range [STARTING_ATTRIBUTE#, ENDING_ATTRIBUTE#] are to be put in the same group GROUP#.
ATTRIBUTE#	Attribute ID number of the elements to be placed in the same group GROUP# (integer).

STARTING_ATTRIBUTE#	First attribute ID number of a sequence of attribute ID numbers that define the group GROUP# (integer).
ENDING_ATTRIBUTE#	Last attribute ID number of a sequence of attribute ID numbers that define the group GROUP# (integer).
N	Sub-command keyword (character) signaling that nodes are to be grouped according to their node identifying number — that is, the node whose identifying number is NODE# or all nodes whose identifying numbers are within the range [STARTING_NODE#, ENDING_NODE#] are to be put in the same group GROUP#.
NODE#	ID number of the node to be placed in the group GROUP# (integer).
STARTING_NODE#	First node ID number of a sequence of node ID numbers that define the group GROUP# (integer).
ENDING_ATTRIBUTE#	Last node ID number of a sequence of node ID numbers that define the group GROUP# (integer).
GROUP#	Group ID number of the constructed group of elements or nodes (integer).

Next: HZEMFILTER, Previous: GROUPS

40 HEAT ZERO ENERGY MODE

Command Statement:	**HZEM**

This command is effective only for quasistatic and transient thermal, thermoelastic, aerothermal, and aerothermoelastic analyses involving thermal zero energy modes. It is used to request the computation of the constant temperature mode and generalized inverse of the conductivity matrix using a physics-based algorithm (rather than the tolerance-based algorithm associated with the TRBM command). More specifically, if the STATICS command is also present in the input file and a direct solver is specified under this command, the generalized inverse of the conductivity matrix is computed in this direct solver by the physics-based method associated with the HZEM command.

HZEM

Next: HEFSB, Previous: HZEM

41 HEAT ZERO ENERGY MODES FILTER

Command Statement:	**HZEMFILTER**

This command is effective for thermal simulations involving thermal zero energy modes. When this command is specified in the input file and the thermal problem contains zero energy modes, a non-trivial projector is constructed. In a static or quasistatic analysis, the transpose of this projector is applied to the (possibly variable) right hand-side to make sure that the external load is self-equilibrated and therefore admissible. In a dynamic analysis: a) this projector is applied to the initial solution (initial displacement *and* velocity fields), and b) at each time-step, the transpose of this projector is applied to the (time-dependent) right-hand side of the equation being solved. In all cases, the net effect is to work with (and output) a solution where the thermal zero energy modes have been filtered out.

The constructed projector can be written as $P = I - Z(Z^T Q Z)^{-1} Z^T Q$ where *I* is the identity matrix, Z is the matrix of the zero energy modes of the thermal system (always computed with the HZEM method), and Q is either the identity matrix for static analysis, or the capacity matrix for quasistatic and dynamic analyses. Hence, in a quasistatic analysis, this command requires the specific heat to be present in the material properties specified under the MATERIAL command.

HZEMFILTER

Next: HEFRS, Previous: HZEMFILTER

42 HYDROELASTIC FLUID/STRUCTURE BOUNDARY

Command Statement:	**HEFSB**

The HEFSB command is used to specify the fluid-structure interface boundary in a hydroelastic eigenvalue problem. The underlying discrete model assumes that each node on this interface is shared by fluid and structural elements. The input format of this command is given below.

Note 1: FEM automatically determines that a hydroelastic eigenvalue problem is to be solved once it finds in the input file the HEFSB command, or the HEFRS command which specifies its free-surface boundary. This problem is solved using the added mass approach. The computed frequencies and structural mode shapes can be output as in a regular structural eigen computation.

HEFSB

FACE#	FACE_TYPE	CONNECTIVITY_NODES

FACE#

Face (or edge in two dimensions) identification number whose type and connectivity are to be specified (integer).

FACE_TYPE

1	2-node line segment. To be used with the appropriate two-dimensional linear elements.
2	3-node line segment. To be used with the appropriate two-dimensional quadratic elements.
3	3-node triangular face. To be used with the appropriate three-dimensional linear tetrahedral elements.
4	4-node quadrilateral face. To be used with the appropriate three-dimensional linear hexahedral elements.
6	6-node triangular face. To be used with the appropriate three-dimensional quadratic tetrahedral elements.
10	Quadrilateral face. To be used with the appropriate higher-order iso-parametric hexahedral elements.
11	Triangular face. To be used with the appropriate higher-order iso-parametric tetrahedral elements.
12	Line segment. To be used with the appropriate higher-order iso-parametric quadrilateral elements.
13	Line segment. To be used with the appropriate higher-order iso-parametric triangular elements.

CONNECTIVITY_NODES	These should be listed in a stacked fashion on a single line. The nodes of low-order faces should be ordered counter-clockwise (when looking from infinity in three dimensions). Those of higher-order faces (type 10–13) should be ordered from left to right and bottom to top using any convenient axis system (when viewed from infinity in three dimensions).

Next: HELMHOLTZ, Previous: HEFSB

43 HYDROELASTIC FREE SURFACE BOUNDARY

Command Statement:	**HEFRS**

The HEFRS command can be used to specify the free surface boundary either in a hydroelastic eigenvalue problem, or in an acoustic or elastoacoustic (fluid-structure interaction) problem where the fluid is assumed to be weightless. At each node located on this free surface, **FEM** automatically applies a zero Dirichlet boundary condition for the fluid potential equation in the first case, and for the scattered pressure in the second case.

The input format of this command is given below.

Note 1: FEM automatically determines that a hydroelastic eigenvalue problem is to be solved once it finds in the input file the HEFRS command, or the HEFSB command which specifies its hydroelastic fluid/structure interface boundary. This problem is solved using the added mass approach. The computed frequencies and structural mode shapes can be output as in a regular structural eigen computation.

```
HEFRS
```

```
FACE#        FACE_TYPE        CONNECTIVITY_NODES
```

FACE#	Face (or edge in two dimensions) identification number whose type and connectivity are to be specified (integer).
FACE_TYPE	
1	2-node line segment. To be used with the appropriate two-dimensional linear elements.
2	3-node line segment. To be used with the appropriate two-dimensional quadratic elements.
3	3-node triangular face. To be used with the appropriate three-dimensional linear tetrahedral elements.
4	4-node quadrilateral face. To be used with the appropriate three-dimensional linear hexahedral elements.
6	6-node triangular face. To be used with the appropriate three-dimensional quadratic tetrahedral elements.
CONNECTIVITY_NODES	These should be listed in a stacked fashion on a single line, and ordered clockwise (when looking from infinity in three dimensions).

Next: HARB, Previous: HEFRS

44 HELMHOLTZ

Command Statement:	**HELM**

The HELM command statement is used to specify a medium-homogeneous, frequency-domain, acoustic (Helmholtz) or elastoacoustic problem and define some of its parameters. Its input format is given below.

Note 1: The time-harmonic form of the solution of the frequency-domain problem considered in the HELM command is $p = Pe^{-i\omega t}$, where P denotes the amplitude of the field p and ω its frequency.

Note 2: The HELM command solves always for the *scattered* pressure p^s. However, one can output the scattered pressure as well as the total pressure.

Note 3: The boundary conditions of the acoustic (Helmholtz) or elastoacoustic problem defined in this command can be set either by the keyword PLANEWAVE or the keyword POINTSOURCE (see below). When either is specified, the REAL_PART_VALUE and IMAGINARY_PART_VALUE of any specified Helmholtz dirichlet boundary condition (see HDIR) are overwritten by $p^s = -p^i$, where p^i is the incident time-harmonic plane wave or spherical wave implied by PLANEWAVE or POINTSOURCE, respectively.

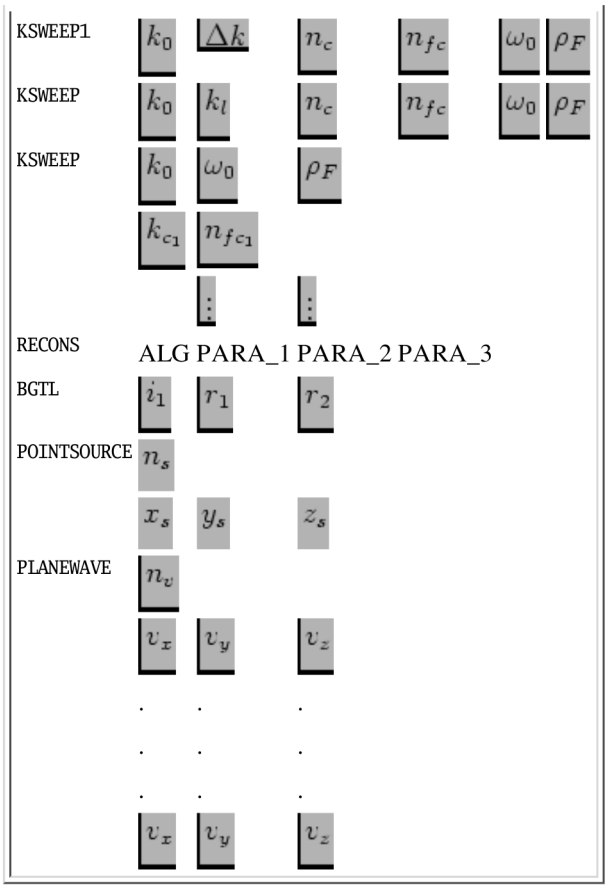
Note 4: The first- and second-order generalized Bayliss-Gunburger-Turkel non reflecting conditions are not implemented for all three-dimensional brick elements.

Note 5: For Helmholtz problems defined in a homogenous (fluid) medium, the user may set MAT_ATT# = 1 in ATTRIBUTES for all of the acoustic elements and not define any material property for this attribute in MATERIAL.

Note 6: For Helmholtz problems defined in a heterogeneous (fluid) medium — and this includes the case where a Perfect Matching Layer is present in the computational domain — the user may define in MATERIAL material properties for the acoustic elements, in which case the wavenumber k and fluid density ρ_F specified under this command are ignored and the wavenumber(s) and fluid densities specified using the FLUMAT sub-command of MATERIAL are used instead.

HELM

k	k	ω	ρ_F
DAMP	a	b	



k — Wavenumber for the fluid system or subsystem (float).

ω — Pulsation for a coupled elastoacoustic problem. Skipping the specification of ω and ρ_F signals a pure acoustic (Helmholtz) problem (float).

ρ_F — Fluid density for a coupled elastoacoustic problem. Skipping the specification of ω and ρ_F signals a pure acoustic (Helmholtz) problem (float).

DAMP — Structured Rayleigh damping for a coupled electroelastic problem.

a — Rayleigh damping stiffness coefficient a (float).

b — Rayleigh damping stiffness coefficient b (float).

KSWEEP1 — This option requests a sweeping on the wavenumber using a one-point scheme to be defined in RECONS. The swept wavenumbers are given by

$$k_0 - n\Delta_k \leq k \leq k_0 + n\Delta_k, \quad -n_{\Delta_k} \leq n \leq n_{\Delta_k}$$

.

k_0 — Wavenumber defining the center of a band (float).

Δ_k — Increment for the wavenumber (float).

n_{Δ_k} — Half the number of increments to be performed (integer).

KSWEEP — This option requests sweeping on the wavenumber using a numerical reconstruction scheme to be defined in RECONS. The swept wavenumbers are defined via a regular or irregular “coarse-grid” wavenumber axis

(characters). In the regular case, the coarse-grid wavenumber points are specified via a wavenumber band [$k_0,\ k_l$] and a number of equally spaced apart frequencies, n_c . Within this wavenumber band, and within each pair of consecutive coarse-grid wavenumber axis points, $n_{fc}-1$ "fine-grid" wavenumber axis points are inserted and the solution of the forced wavenumber response problem is reconstructed at each of thes fine-grid wavenumber axix points. In the irregular case, the first wavenumber point k_0 is specified on the first line after the keyword KSWEEP. Next, on a separate line, a coarse-grid wavenumber point k_{c_1} can be specified together with an integer number n_{fc_1} specifying that $n_{fc}-1$ equally spaced "fine-grid" wavenumber axis points are to be generated in the interval $[k_0,\ k_{c_1}]$. The solution of the forced wavenumber response problem will be reconstructed at each of these fine-grid wavenumber axis points. This separate line can be repeated as many times as desired, with the understanding that the implied wavenumber intervals are $[k_{c_1},\ k_{c_2}]$, then $[k_{c_2},\ k_{c_3}]$ etc. Clearly, this command is best exploited if the k_{c_i} values are input in a monotonic sequence of increasing or decreasing numbers.

RECONS — Defines the reconstruction strategy in the case of a wavenumber sweep and sets its parameters.

ALG — Reconstruction algorithm (characters). Three such algorithms are available and listed below.

Taylor — Taylor series algorithm (characters).

Pade — Conventional Pade algorithm (characters).

PadeLanczos — Pade algorithm based on a Lanczos procedure (characters). This version of the Pade reconstruction algorithm is less prone to ill-conditioning and therefore is more performing.

PARA_1 — First parameter of the reconstruction algorithm. For a Taylor series, it specifies the order of the series. For a Pade series, it specifies the number of points on the coarse-grid wavenumber axis to be used at-one-time to reconstruct the solution.

PARA_2 — Second parameter of the reconstruction algorithm that is relevant only for the case of a Pade-based reconstruction algorithm. It specifies the order of the numerator of the rational function (L of [L/M]) (integer).

PARA_3 — Third parameter of the reconstruction algorithm that is relevant only for the case of a Pade-based reconstruction algorithm. It specifies the order of the denominator of the rational function (M of [L/M]) (integer).

BGTL

i_1 — Order (0, 1, or 2) of the generalized Bayliss-Gunburger-Turkel non reflecting condition to be applied to the artificial boundary Σ specified under the HARB command (integer).

r_1 — This real number (float) is to be used only when the artificial boundary Σ (see HARB) is supposed to be either a sphere or an ellipsoid, and some geometric approximations generated by FEM are to be overwritten by values that can in that case be evaluated exactly. Otherwise, r_1 should simply not be input. For a shpere, the curvature of Σ , which is otherwise automatically approximated by FEM is overwritten by $1/r_1$. Similarly, for an ellipsoid of the form $(x/a)^2+(y/b)^2+(z/c)^2=1$, $r_1=a$.

r_2 — This real number (float) is to be used only when the artificial boundary Σ (see HARB) is supposed to be an ellipsoid, and some geometric approximations generated by FEM are to be overwritten by values that

can in that case be evaluated exactly. Otherwise, r_2 should simply not be input. For an ellipsoid of the form $(x/a)^2 + (y/b)^2 + (z/c)^2 = 1$, $r_2 = b = c$.

POINTSOURCE — Keyword indicating that the Dirichlet and/or Neumann boundary conditions (see HDIR and HDNB) of the acoustic (Helmholtz) or elastoacoustic problem defined by this command are associated with a time-harmonic spherical wave of the form $\frac{1}{4\pi}\frac{e^{ikr}}{r}$, where $r = |X - X_s|$, and propagated by a source located at the point X_s (character). Multiple spherical waves and sources can be specified under this command leading to multiple acoustic or elastoacoustic analyses. Hence, this line should be followed by a line specifying the number of sources and spherical waves, n_s, and n_s lines specifying for each wave its source point X_s. When used to define Neumann boundary conditions (see HDNB), this option is available only for the faces of type 10 and 11 (see HDNB).

n_s — Number of sources propagating spherical waves (integer).

x_s — x coordinate of the location of a point source (float).

y_s — y coordinate of the location of a point source (float).

z_s — z coordinate of the location of a point source (float).

PLANEWAVE — Keyword indicating that the Dirichlet and/or Neumann boundary conditions (see HDIR and HDNB) of the acoustic (Helmholtz) or elastoacoustic problem defined by this command are associated with an incident time-harmonic plane wave of the form $e^{ik\vec{x}.\vec{v}}$ (character). Multiple such waves can be specified under this command leading to multiple acoustic or elastoacoustic analyses. Hence, this line should be followed by a line specifying the number of plane waves to be considered, n_v, and n_v lines specifying for each wave its direction of propagation v.

n_v — Number of incident planar waves (integer).

v_x — Normalized x component of the direction of a planar wave (float).

v_y — Normalized y component of the direction of a planar wave (float).

v_z — Normalized z component of the direction of a planar wave (float).

Next: HDIR, Previous: HELMHOLTZ

45 HELMHOLTZ ARTIFICIAL BOUNDARY *S*

Command Statement:	**HARB**

The HARB command statement is used to specify the artificial boundary Σ on which the absorbing condition specified in the HELMHOLTZ command (see HELMHOLTZ) is to be applied. The input format is given below.

Note 1: In general, the absorbing condition in frequency domain computations is applied to the solution variable. Since in frequency

domain computation it does not make sense to absorb the incident field, the solution variable in FEM is in this case the scattered field.

HARB

FACE# FACETYPE CONNECTIVITY_NODES

FACE#	Face (or edge in two dimensions) id number whose type and connectivity are to be specified (integer). In practice, this id number is ignored by FEM.
FACE_TYPE	
1	2-node line segment. To be used with two-dimensional linear elements.
2	3-node line segment. To be used with two-dimensional quadratic elements.
3	3-node triangular face. To be used with three-dimensional linear tetrahedral element.
4	4-node quad face. To be used with three-dimensional linear hexahedral element.
6	6-node triangular face. To be used with three-dimensional quadratic tetrahedral element.
10	4, 9, 16 or 25-node quadrilaterial face. To be used with three-dimensional full isoparametric hexahedral elements.
11	3, 6, 10 or 15-node triangular face. To be used with three-dimensional full isoparametric tetrahedral elements.
12	2, 3, 4 or 5-node line segment. To be used with two-dimensional full isoparametric quadrilateral elements.
13	2, 3 or 4-node line segment. To be used with two-dimensional full isoparametric triangular elements.
CONNECTIVITY_NODES	These should be listed in a stacked fashion on a single line, and numbered clockwise (when looking from outside in three dimensions).

Next: HDNB, Previous: HARB

46 HELMHOLTZ DIRICHLET BOUNDARY CONDITIONS *S*

Command Statement: **HDIR**

The HDIR command statement is used to specify *nodal* Dirichlet boundary conditions for a frequency-domain acoustic scattering (Helmholtz) problem. The input format is given below.

Note 1: When the keyword PLANEWAVE or POINTSOURCE is specified under HELMHOLTZ, it overwrites the REAL_PART_VALUE and IMAGINARY_PART_VALUE (see below) by $p^s = -p^i$, where p^i is an incident time-harmonic plane wave specified by PLANEWAVE, or an incident time-harmonic spherical wave propagating from a source located at a point specified by POINTSOURCE.

HDIR

NODE#	DOF#	REAL_PART_VALUE	IMAGINARY_PART_VALUE

NODE#	Node number where the Dirichlet boundary condition is specified (integer).
DOF#	Degree of freedom local number where the boundary condition is specified (integer). This number should be set to 8.
REAL_PART_VALUE	Real part of the value of the specified boundary condition (float).
IMAGINARY_PART_VALUE	Imaginary part of the value of the specified boundary condition (float).

Next: HNEU, Previous: HDIR

47 HELMHOLTZ DISTRIBUTED NEUMANN BOUNDARY CONDITION *S*

Command Statement:	**HDNB**

The HDNB command statement can be used to specify on all or segments of the surface of a scatterer a *distributed* Neumann boundary condition of the form $\frac{\partial p^s}{\partial n} = -\frac{\partial p^i}{\partial n}$, where the superscripts s and i designate scattered and incident quantities, respectively. The incident time-harmonic wave can be a time-harmonic plane wave specified by PLANEWAVE, or a time-harmonic spherical wave propagating from a source located at a point specified by POINTSOURCE in HELMHOLTZ. Note that for problems involving a nonhomogenous fluid, HDNB can only be used to define a surface that is submerged entirely in one and only one fluid.

The input format of this command is given below.

HDNB

FACE#	FACE_TYPE	CONNECTIVITY_NODES

FACE#	Face (or edge in two dimensions) id number whose type and connectivity are to be specified (integer). In practice, this id number is ignored by FEM.
FACE_TYPE	
1	2-node line segment. To be used with two-dimensional linear elements.
2	3-node line segment. To be used with two-dimensional quadratic elements.
3	3-node triangular face. To be used with three-dimensional linear tetrahedral elements.
4	4-node quad face. To be used with three-dimensional linear hexahedral elements.
6	6-node triangular face. To be used with three-dimensional quadratic tetrahedral elements.
10	4, 9, 16 or 25-node quadrilaterial face. To be used with three-dimensional full isoparametric hexahedral elements.

11	3, 6, 10 or 15-node triangular face. To be used with three-dimensional full isoparametric tetrahedral elements.
12	2, 3, 4 or 5-node line segment. To be used with two-dimensional full isoparametric quadrilateral elements.
13	2, 3 or 4-node line segment. To be used with two-dimensional full isoparametric triangular elements.
CONNECTIVITY_NODES	These should be listed in a stacked fashion on a single line, and numbered counter clockwise (when looking from infinity in three dimensions).

Next: HSCB, Previous: HDNB

48 HELMHOLTZ NEUMANN BOUNDARY CONDITIONS *S*

Command Statement: **HNEU**

The HNEU command statement is used to specify the *nodal* Neumann boundary conditions for a frequency-domain acoustic scattering (Helmholtz) problem. The input format is given below.

```
HNEU
```

```
HNEU        REAL_PART_VALUE        IMAGINARY_PART_VALUE
```

NODE#	Node number where the Neumann boundary condition is specified (integer).
REAL_PART_VALUE	Real part of the value of the specified boundary condition (float).
IMAGINARY_PART_VALUE	Imaginary part of the value of the specified boundary condition (float).

Next: HWIB, Previous: HNEU

49 HELMHOLTZ SCATTERER BOUNDARY *S*

Command Statement: **HSCB**

The HSCB command statement is used to specify the surface of a scatterer. It is used to compute the farfield pattern of a frequency-domain acoustic solution, and therefore is necessary for the output of that farfield pattern (see OUTPUT).

The input format of this command is given below.

```
HSCB
```

FACE#	FACE_TYPE	CONNECTIVITY_NODES

FACE#	Face (or edge in two dimensions) id number whose type and connectivity are to be specified (integer). In practice, this id number is ignored by FEM.
FACE_TYPE	
1	2-node line segment. To be used with two-dimensional linear elements.
2	3-node line segment. To be used with two-dimensional quadratic elements.
3	3-node triangular face. To be used with three-dimensional linear tetrahedral elements.
4	4-node quad face. To be used with three-dimensional linear hexahedral elements.
6	6-node triangular face. To be used with three-dimensional quadratic tetrahedral elements.
10	4, 9, 16 or 25-node quadrilaterial face. To be used with three-dimensional full isoparametric hexahedral elements.
11	3, 6, 10 or 15-node triangular. To be used with three-dimensional full isoparametric tetrahedral elements.
12	2, 3, 4 or 5-node line segment. To be used with two-dimensional full isoparametric quadrilateral elements.
13	2, 3 or 4-node line segment. To be used with two-dimensional full isoparametric triangular elements.
CONNECTIVITY_NODES	These should be listed in a stacked fashion on a single line, and numbered counter clockwise (when looking from outside in three dimensions).

Next: HLMPC, Previous: HSCB

50 HELMHOLTZ WET INTERFACE BOUNDARY *S*

Command Statement:	**HWIB**

The HWIB command statement is used to describe the wet interface boundary for coupled elasto-acoutic frequency response analysis using fluid and structure meshes with matching discrete interfaces. The input format is given below.

Note 1: This is an alternative way to specify a **matching** fluid/structure interface using the same input format as that for the HARB and HDNB commands (see HARB and see HDNB) but is supported only for face types 10, 11, 12 and 13.

Note 2: In this case, the capabilities of the ACME library are not used to generate the fluid-structure coupling coefficients.

HWIB

FACE#	FACETYPE	CONNECTIVITY_NODES

FACE#	Face (or edge in two dimensions) id number whose type and connectivity are to be specified (integer). In practice, this id number is ignored by FEM.
FACETYPE	

10	4, 9, 16 or 25-node quadrilaterial face. To be used with three-dimensional full isoparametric hexahedral elements.
11	3, 6, 10 or 15-node triangular. To be used with three-dimensional full isoparametric tetrahedral elements.
12	2, 3, 4 or 5-node line segment. To be used with two-dimensional full isoparametric quadrilateral elements.
13	2, 3 or 4-node line segment. To be used with two-dimensional full isoparametric triangular elements.
CONNECTIVITY_NODES	These should be listed in a stacked fashion on a single line, and numbered as shown in the following examples: 9-noded quad: 7 8 9 4 5 6 1 2 3 6-noded triangle: 6 4 5 1 2 3 etc.

Next: KIRLOC, Previous: HWIB

51 HELMHOLTZ LINEAR MULTIPOINT CONSTRAINTS

Command Statement:	**HLMPC**

The HLMPC command statement is used to specify a set of *complex* linear multipoint constraint equations of the form

$$\sum_{j=1}^{j=n} c_{ij} u_j = r_i, \quad i = 1, 2, \ldots$$

in a frequency-domain acoustic model. There is no limitation on the number of multipoint constraints, or number of degrees of freedom related by the same constraint equation. The format of this command statement is as follows.

Note 1: The Lagrange multiplier method for enforcing the constraints associated with this command is supported only by the FETI-DP family of solvers, the GMRES solver, and the SPOOLES and MUMPS direct sparse solvers (see STATICS).

HLMPC

CONSTRAINT#	RHS	CONSTRAINT_METHOD	
NODE#	DOF#	REAL-PART-COEFF	IMAGINARY-PART-COEFF
.			

```
.
NODE#        DOF#        REAL-PART-COEFF    IMAGINARY-
                                            PART-COEFF
```

CONSTRAINT#	This corresponds to the constraint equation number *i* (integer).
RHS	This is the right-hand side r_i of the *i*-th constraint equation (float). It can have both a real and an imaginary part, in which case the real part is specified first and followed by the imaginary part.
CONSTRAINT_METHOD	This is the method for enforcing the constraint (characters). The default method is set in CONSTRAINTS and used whenever this entry is omitted.
multipliers	The Lagrange multiplier method.
elimination	The elimination method.
penalty [beta]	The penalty method. The parameter beta should be a large positive number, typically of the order of 10^8 (no default value is provided).
NODE#	This is the number of the node contributing the coefficient c_{ij} of the *i*-th constraint equation (integer).
DOF#	This is the local number of the degree of freedom at the node specified above contributing the coefficient c_{ij} of the *i*-th constraint equation (integer).
REAL_PART_VALUE	This is the real part of the coefficient c_{ij} of the *i*-th constraint equation (float).
IMAGINARY_PART_VALUE	This is the imaginary part of the coefficient c_{ij} of the *i*-th constraint equation (float).

Next: IMPEDANCE, Previous: HLMPC

52 HELMHOLTZ LOCATIONS WHERE TO COMPUTE THE KIRCHHOFF INTEGRAL

Command Statement:	**KIRLOC**

When KIRCHHOFF is selected as an OUTPUT result and the surface of the scatterer is defined using the command HSCB, this command should be used to specify the locations of the points where the (far-field) solution of the HELMHOLTZ problem to be solved is to be evaluated using the Kirchhoff integral and outputted. These points do not need to be nodes of the mesh. Usually, but not necessarily, they are outside the computational domain, in the far-field. The keyword KIRLOC should be followed by as many lines as there are points where it is desired to evaluate the solution of the frequency-domain acoustic or elastoacoustic problem defined in HELMHOLTZ. Each line contains the coordinates of such a point.

The syntax of this command is given below.

```
KIRLOC
```

X-ORDINATE	Y-ORDINATE	Z-ORDINATE

X-ORDINATE X-ordinate of a point where to evaluate the solution of the HELMHOLTZ problem to be solved using the Kirchhoff integral (real).

Y-ORDINATE Y-ordinate of a point where to evaluate the solution of the HELMHOLTZ problem to be solved using the Kirchhoff integral (real).

Z-ORDINATE Z-ordinate of a point where to evaluate the solution of the HELMHOLTZ problem to be solved using the Kirchhoff integral (real).

Next: IACCELERATIONS, Previous: KIRLOC

53 IMPEDANCE

Command Statement:	**IMPEDANCE**

The IMPEDANCE command statement can be used for three purposes: 1) to signal that the problem to be solved is a forced frequency response structural dynamics problem of the form

$$(K - \omega^2 M - i\omega D)u = f$$

where K is a stiffness matrix, M is a mass matrix, $D = aK + bM$ is a Rayleigh damping matrix, $\omega^2 = 4\pi^2 f^2$ with f denoting the forced frequency which can be varied in a specified interval, u is the amplitude of the time-harmonic solution $v = ue^{-i\omega t}$, and g is the amplitude of the time-harmonic forcing function $h = ge^{-i\omega t}$, 2) to specify a shift σ to be applied to the stiffness matrix so that it is replaced in any dynamic analysis specified by any other command such as EIGEN or other by

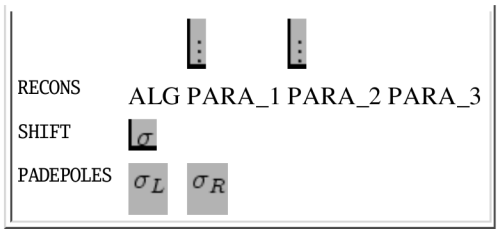
$$\tilde{K} = K - \sigma M$$

, and 3) to identify within a specified range of interest the eigenvalues missed by a previous eigenvalue computation and their corresponding eigenvectors.

The input format of this command is given below.

IMPEDANCE

FREQ	f				
DAMP	a	b			
FREQSWEEP1	f_0	Δf	n_{Δ_f}		
FREQSWEEP	f_0	f_l	n_c	n_{fc}	
FREQSWEEP	f_0				
	f_{c_1}	n_{fc_1}			



FREQ

f Value of the forced frequency (float). The corresponding value of ω^2 is $4\pi^2 f^2$.

DAMP

a Rayleigh damping stiffness coefficient a (float).

b Rayleigh damping stiffness coefficient b (float).

FREQSWEEP1 This option requests sweeping on the frequency using a one-point scheme to be specified in RECONS. The swept frequencies are given by

$$f_0 - n\Delta_f \le f \le f_0 + n\Delta_f, \quad -n_{\Delta_f} \le n \le n_{\Delta_f}$$

.

f_0 Frequency defining the center of a band (float).

Δ_f Increment for the frequency (float).

n_{Δ_f} Half the number of increments to be performed (integer).

FREQSWEEP This option requests sweeping on the frequencies defined by refining a regular or irregular “coarse-grid” frequency axis (characters). In the regular case, the coarse-grid frequency points are specified via a frequency band [$f_0,\ f_l$] and a number of equally spaced apart frequencies, n_c . Within this frequency band, and within each pair of consecutive coarse-grid frequency axis points, $n_{fc} - 1$ “fine-grid” frequency axis points are inserted and the solution of the forced frequency response problem is reconstructed at each of thes fine-grid frequency axix points. In the irregular case, the first frequency point f_0 is specified on the first line after the keyword FREQSWEEP. Next, on a separate line, a coarse-grid frequency point f_{c_1} can be specified together with an integer number n_{fc_1} specifying that $n_{fc} - 1$ equally spaced “fine-grid” frequency axis points are to be generated in the interval $[f_0,\ f_{c_1}]$. The solution of the forced frequency response problem will be reconstructed at each of these fine-grid frequency axis points. This separate line can be repeated as many times as desired, with the understanding that the implied frequency intervals are $[f_{c_1},\ f_{c_2}]$, then $[f_{c_2},\ f_{c_3}]$ etc. Clearly, this command is best exploited if the f_{c_i} values are input in a monotonic sequence of increasing or decreasing numbers.

RECONS Defines reconstruction strategy in the case of a frequency sweep and sets its parameters.

ALG Reconstruction algorithm (characters). Three such algorithms are available and listed below.

Taylor Taylor series algorithm (characters).

Pade Conventional (multipoint) Pade algorithm (characters).

PadeLanczos (multipoint) Pade algorithm based on a Lanczos procedure (characters). This version of the Pade reconstruction algorithm is less prone to ill-conditioning and therefore is more performing.

PARA_1 First parameter of the reconstruction algorithm. For a Taylor series, it specifies the order of the series. For a Pade series, it specifies the number of points on the coarse-grid frequency axis to be used at-one-time to reconstruct the solution.

PARA_2 Second parameter of the reconstruction algorithm that is relevant only for the case of a Pade-based reconstruction algorithm. It specifies the order of the numerator of the rational function (L of [L/M]) (integer).

PARA_3 Third parameter of the reconstruction algorithm that is relevant only for the case of a Pade-based reconstruction algorithm. It specifies the order of the denominator of the rational function (M of [L/M]) (integer).

SHIFT Specifies a shift σ for the stiffness matrix (real).

PADEPOLES Can be used only with ALG = PadeLanczos and needs input from the FREQSWEEP sub-command (see above). Also, requires the presence in the input file of the MODE command in order to retrieve the EIGENMODES file or files placed in the execution path. In this case, the Pad\'e-based approximation of a rational transfer function whose poles are exactly the eigenvalues of the symmetric pencil $(\mathbf{K}, \mathbf{M})$ is computed and the poles of this approximation are exploited to identify, in a specified range of interest $[\sigma_L, \sigma_R]$, the eigenvalues that may have been missed by a previous eigen computation in which the MODE command was used to store the results in the EIGENMODES file or files (characters). The points $\sigma_i \in [\sigma_L, \sigma_R]$ needed by the multipoint Pade expansion should be generated with the FREQSWEEP sub-command and σ_i playing the role of f_i, $f_0 = \sigma_L$, $f_l = \sigma_R$ (and for example, $n_{fc} = 0$). The parameters of this expansion should be specified in PARA_1, PARA_2 and PARA_3 (see above). At the end of the computation, **FEM** outputs on the screen the poles of the Pade rational function in the specified range $[\sigma_L, \sigma_R]$, excluding those poles corresponding to the modes read from the EIGENMODE file or files. The specified range $[\sigma_L, \sigma_R]$ can be narrower than that of the eigenvalues read in the EIGENMODE file or files. However, if it is wider, some of the poles of the Pade rational function may, in some cases, not correspond to missed eigenvalues. In any case, the output of the eigenvectors associated with the poles or missed eigenvalues is not currently implemented in **FEM**.

σ_L Lower bound of an eigenvalue interval of interest (real).

σ_L Upper bound of an eigenvalue interval of interest (real).

Next: IDISPLACEMENTS, Previous: IMPEDANCE

54 INITIAL ACCELERATIONS (Not Supported Yet)

Command Statement:	**IACCELERATIONS**

The IACCELERATIONS command statement is used to specify a nodal acceleration type of initial conditions. Each node can have up to six degrees of freedom.

IACCELERATIONS

```
NODE#        DOF#        VALUE
```

NODE#	Node number where the initial acceleration is specified (integer).
DOF#	Degree of freedom local number where the initial acceleration is specified (integer).
VALUE	Value of the specified initial acceleration (float).

Next: IDISP6, Previous: IACCELERATIONS

55 INITIAL DISPLACEMENTS *S*

Command Statement:	**IDISPLACEMENTS**

For a structural analysis, the IDISPLACEMENTS command statement can be used to specify a nodal (or generalized coordinate) displacement type of initial conditions (in this case, each node can have up to six degrees of freedom). For a time-domain acoustic analysis, this command statement can be used to specify initial nodal values of the solution by setting DOF# to 8 (see below).

Note 1: If theGEPS command is not specified, and both DISP and IDISP6 commands are present in the input file, **FEM** selects the IDISP6 command to initialize the displacement field.

Note 2: If the GEPS command is present in the input file, then **FEM** uses the content of the command IDISP6 to construct the geometric stiffness, and the content of IDISPLACEMENTS to initialize the displacement field. If IDISPLACEMENTS is not specified in the input file, then **FEM** uses the command IDISP6 to perform both tasks of constructing the geometric stiffness and initializing the displacement field.

When working with the usual basis, the format of this command is as follows.

```
IDISPLACEMENTS
```

or

```
IDISPLACEMENTS ZERO
```

```
NODE#        DOF#        VALUE
```

NODE#	Node number where the initial displacement or acoustic field is specified (integer).
DOF#	Degree of freedom local number where the initial displacement or acoustic field is specified (integer).
VALUE	Value of the specified initial displacement or acoustic field (real).

ZERO	This initializes the entire intital displacement or acoustic field to zero. No other IDISPLACEMENTS data has to be entered when this option is selected (characters).

When working in the basis of generalized coordinates (or reduced basis), the format of this command is as follows.

```
IDISPLACEMENTS
MODAL
MODE#          VALUE
```

MODAL	This keyword, completely spelled out, can be used either (1) to input the initial conditions in the generalized coordinates (or reduced) system, whether a modal or direct time-integration analysis is requested in the DYNAMICS command, or (2) to superpose to the initial conditions specified in the global coordinates system additional initial conditions formulated in the generalized coordinates (or reduced) system (characters). In the second case, IDISPLACEMENTS and/or IVELOCITIES should be specified only once and the keyword MODAL can be inserted either before or after the set of data associated with the initial conditions specified in the global coordinates. This option can also be used to perform a "Ping-Pong" analysis (AERO). In both cases, the READMODE command statement must also be present in the input file to provide the eigenmodes to be used.
MODE#	Mode id number for which the initial condition is specified (integer).
VALUE	Value of the specified initial condition (real).

Next: IDISP6PITA, Previous: IDISPLACEMENTS

56 INITIAL DISPLACEMENT 6 COLUMNS (IDISP6 completely spelled out) *S*

Command Statement: **IDISP6**

The IDISP6 command statement is used to specify a nodal displacement type of initial conditions in a different format than that of the IDISPLACEMENTS command. This format is specified below.

Note 1: If the GEPS command is not specified, and both IDISPLACEMENTS and IDISP6 commands are present in the input file, **FEM** selects the IDISP6 command to initialize the displacement field.

Note 2: If the GEPS command is present in the input file, then **FEM** uses the content of the command IDISP6 to construct the geometric stiffness, and the content of IDISPLACEMENTS to initialize the displacement field. If IDISPLACEMENTS is not specified in the input file, then **FEM** uses the command IDISP6 to perform both tasks of constructing the geometric stiffness and initializing the displacement field.

Note 3: Modal initial condition is not supported under IDISP6 because it is not relevant.

IDISP6

NODE#	VAL_DOF1	VAL_DOF2	VAL_DOF3	VAL_DOF4	VAL_DOF5	VAL_DOF6

	Amplification facteur that multiplies each VAL_DOFi value for each node (real).
NODE#	Node number where the initial displacement is specified (integer).
VAL_DOFi#	Value of the specified initial displacement for the i-th degree of freedom (real).

Next: IVEL6PITA, Previous: IDISP6

57 INITIAL OR SEED DISPLACEMENT FOR PITA (IDISP6PITA completely spelled out)

Command Statement:	**IDISP6PITA**

The IDISP6PITA command statement can be used to specify the time-slices' initial or seed displacements for the PITA methodology (see PITA), thus overriding the alternative approach consisting of generating this information by applying the basic time-integrator on the coarse time-grid. It is applicalble in both linear and nonlinear settings. Its syntax is similar to that of the IDISP6 (see IDISP6) command except that it can introduce up to as many sets of initial displacements as there are time-slices, one set per time-slice. Each set of initial displacements must be input after the previous one, starting with that corresponding to the first time-slice and continuing in consecutive order. All those time-slices for which a set of seed displacement values is not specified are initialized by the aforementioned alternative approach.

IDISP6PITA	NUMSLICES

NODE#	VAL_DOF1	VAL_DOF2	VAL_DOF3	VAL_DOF4	VAL_DOF5	VAL_DOF6

NUMSLICES	Number of consecutive time-slices for which seed displacements are specified (integer).
NODE#	Node number where the seed displacement is specified (integer).
VAL_DOFi#	Value of the seed displacement for the i-th degree of freedom at the beginning of a time-slice (float).

Next: ITEMPERATURES, Previous: IDISP6PITA

58 INITIAL OR SEED VELOCITY FOR PITA (IVEL6PITA completely spelled out)

Command Statement:	**IVEL6PITA**

The IVEL6PITA command statement can be used to specify the time-slices' initial or seed velocities for the PITA methodology (see PITA), thus overriding the alternative approach consisting of generating this information by applying the basic time-integrator on the coarse time-grid. It is applicalble in both linear and nonlinear settings. Its syntax is similar to that of the IDISP6 (see IDISP6) command except that it can introduce up to as many sets of initial velocities as there are time-slices, one set per time-slice. Each set of initial velocities must be input after the previous one, starting with that corresponding to the first time-slice and continuing in consecutive order. All those time-slices for which a set of seed velocity values is not specified are initialized by the aforementioned alternative approach.

```
IVEL6PITA        NUMSLICES
```

```
NODE#    VAL_DOF1    VAL_DOF2    VAL_DOF3    VAL_DOF4    VAL_DOF5    VAL_DOF6
```

NUMSLICES	Number of consecutive time-slices for which seed velocities are specified (integer).
NODE#	Node number where the seed velocity is specified (integer).
VAL_DOFi#	Value of the seed velocity for the i-th degree of freedom at the beginning of a time-slice (float).

Next: IVELOCITIES, Previous: IVEL6PITA

59 INITIAL TEMPERATURES *S*

Command Statement:	**ITEMPERATURES**

The ITEMPERATURES command statement is used to specify a nodal displacement type of initial conditions. Each node can have one degree of freedom.

```
ITEMPERATURES
```

```
NODE        VALUE
```

NODE#	Node number where the initial temperature is specified (integer).
VALUE	Value of the specified initial temperature (float).

Next: INPC, Previous: ITEMPERATURES

60 INITIAL VELOCITIES *S*

Command Statement:	**IVELOCITIES**

The IVELOCITIES command statement is used to specify a nodal velocity type of initial conditions. Each node can have up to six degrees of freedom. For a time-domain acoustic analysis, this command statement can be used to specify nodal values of the initial time-derivative of the solution by setting DOF# to 8 (see below).

```
IVELOCITIES
```

```
NODE#          DOF#          VALUE
```

NODE#	Node number where the initial velocity is specified (integer).
DOF#	Degree of freedom local number where the initial velocity is specified (integer).
VALUE	Value of the specified initial temperature (float).

When working in the basis of generalized coordinates (or reduced basis), the format of this command is as follows.

```
IVELOCITIES
MODAL
MODE#          VALUE
```

MODAL	This keyword, completely spelled out, can be used either (1) to input the initial solution time-derivatives in the generalized coordinates (or reduced) system, whether a modal or direct time-integration analysis is requested in the DYNAMICS command, or (2) to superpose to the initial velocity conditions specified in the global coordinates system additional initial velocity conditions formulated in the generalized coordinates (or reduced) system (characters). In the second case, IVELOCITIES should be specified only once and the keyword MODAL can be inserted either before or after the set of data associated with the initial velocity conditions specified in the global coordinates system. The READMODE command statement must also be present in the input file to provide the eigenmodes to be used.
MODE#	Mode id number for which the initial condition is specified (integer).
VALUE	Value of the specified initial condition (real).

Next: MATERIAL, Previous: IVELOCITIES

61 INTRUSIVE POLYNOMIAL CHAOS

Command Statement:	**INPC**

The INPC command signals to the FEM code to perform an intrusive non deterministic analysis using the Polynomial Chaos

representation of the solution. It requires that the input file also contains the GROUP, RANDOM, and STATICS commands and their respective data.

Note 1: Currently, this command is limited to linear, static, structural analysis.

Note 2: Currently, this command is supported only by the PCG, BCG, and CR iterative solvers (see STATICS).

The input format of this command is given below.

INPC

DEG_OUT

DEG_OUT	Degree of the polynomial chaos representation of the solution (integer).

Next: LMPC, Previous: INPC

62 LINEAR MATERIAL PROPERTIES *S*

Command Statement:	**MATERIAL**

The MATERIAL command statement is used to signal that the following data lines correspond **and/or point to**: (1) definitions of materials and geometric properties for materially linear elements, and/or geometric properties for materially nonlinear elements, and (2) requests for specific methods for enforcing the constraints associated with joint and rigid elements (see CONSTRAINTS). Several input formats are possible for this command and are described below.

Note 1: Even for a given input format, the interpretation of a data entry can change from one element type to another. For example for the first input format, the main interpretation of some data entries changes when the element type is that of a lumped torsional spring, a torsional spring connector, a linear spring connector, a Timoshenko beam, a thermal element, a rigid translational link, or a rigid rotational link.

Note 2: For composite elements, this command is not required unless stress and/or strain output is requested. Even in this case, only the total thickness of the element must be specified under this command. Dummy or zero values can be entered for the other properties as their real values should be specified under the COMPOSITE command.

MATERIAL

MID	A	E	ν	ρ	h	k	t	P	Ta	c_p	w	Ixx	Iyy	Izz	ymin	ymax	zmin	zmax

MID	The material id number.
A	Cross sectional area for trusses, beams, and lineal thermal elements (float).
E	Young's Modulus for all element types (float). However, this field can also be filled by a **negative integer** in which case it means that Young's Modulus is a function of temperature and is to be determined from the YMIT and TEMPERATURES commands. In the latter case, the integer specified here points to the id of the Young's modulus-temperature table in the YMIT command (negative integer).
ν	Poisson's ratio for all element types except trusses (float).
ρ	Mass density per unit volume except for element type 20 (composite or orthotropic shell element) when defined using the COEF option, in which case it is mass density per unit area (float).
h	Heat convection coefficient (only for boundary convection elements: type 47, 48, and 49 (float)).
k	Heat conduction coefficient (float).
t	Thickness for plate and shell elements (float).
P	Perimeter/circumference area for thermal elements (float), or the depth of the boundary where convection occurs when element type 47 is used to model non-lateral boundary convection (float).
Ta	Reference temperature (Absolute) of the element (float). **WARNING: if this value is different from the nodal temperature of that element or this value is different from zero and no nodal temperature is specified, it creates a thermal loading even in a pure structural analysis.**
c_p	Specific heat coefficient in a thermal analysis, or Rayleigh damping stiffness coefficient in a structural dynamic analysis (float). When interpreted as a Rayleigh damping stiffness coefficient, this value overrides for this MID any value of the Rayleigh damping coefficient specified under DYNAMICS.
w	Coefficient of thermal expansion in a thermal analysis, or Rayleigh damping mass coefficient in a structural dynamic analysis (float). In a thermal analysis, this field can also be filled by a **negative integer** meaning that the coefficient of thermal expansion is a function of temperature and is to be determined from the TETT and TEMPERATURES commands. In this case, the integer specified here points to the id of the coefficient of thermal expansion-temperature table in the TETT command (negative integer). In a structural dynamic analysis, this value overrides for this MID any value of the Rayleigh damping mass coefficient specified under DYNAMICS.
Ixx/ss	For a mechanical analysis (MECH under the DYNAMICS command), this is the cross-sectional moment of inertia about the local and centroidal x-axis. For an acoustic analysis in the time-domain (ACOU under the DYNAMICS command), this is the speed of sound in the medium represented by the attribute number.
Iyy	Cross-sectional moment of inertia about the local and centroidal *principal* y-axis.
Izz	Cross-sectional moment of inertia about the local and centroidal *principal* z-axis.
ymin	Negative local y-coordinate of the bottom fiber of a beam cross section.
ymax	Positive local y-coordinate of the top fiber of a beam cross section.
zmin	Negative local z-coordinate of the top fiber of a beam cross section.
zmax	Positive local z-coordinate of the top fiber of a beam cross section.

If the element is a lumped torsional spring (eltyp = 11), a linear spring connector (eltyp = 21), or a torsional spring connector (eltyp = 22), the material properties are defined as follows.

MID	Kx	Ky	Kz	lx1	ly1	lz1	lx2	ly2	lz2	lx3	ly3	lz3

MID	The material id number from element attribute table.
Kx	Torsional/Translational spring constant along local x-axis (float).
Ky	Torsional/Translational spring constant along local y-axis (float).
Kz	Torsional/Translational spring constant along local z-axis (float).
lx1 ~ ly1 ~ lz1	The first axis of the local frame expressed in the global frame (floats).
lx2 ~ ly2 ~ lz2	The first axis of the local frame expressed in the global frame (floats).
lx3 ~ ly3 ~ lz3	The first axis of the local frame expressed in the global frame (floats).

If the element is a Timoshenko beam (eltyp = 7), the material properties are defined as follows.

MID	A	E	ν	ρ	α_Y	α_Z	C1	P	Ta	c_p	w	Ixx	Iyy	Izz	ymin	ymax	zmin	zmax

MID	The material id number from element attribute table.
A	Cross sectional area (float).
E	Young's Modulus (float).
ν	Poisson's ratio (float).
ρ	Mass density per unit volume (float).
α_Y	Shear deflection constant associated with Iyy (float).
α_Z	Shear deflection constant associated with Izz (float).
C1	Non-uniform torsion constant (float).
P	Not Applicable
Ta	Not Applicable
c_p	Not Applicable
w	Not Applicable
Ixx	Cross-sectional moment of inertia about the local and centroidal x-axis
Iyy	Cross-sectional moment of inertia about the local and centroidal *principal* y-axis.
Izz	Cross-sectional moment of inertia about the local and centroidal *principal* z-axis.
ymin	Negative local y-coordinate of the bottom fiber of a beam cross section.
ymax	Positive local y-coordinate of the top fiber of a beam cross section.
zmin	Negative local z-coordinate of the top fiber of a beam cross section.
zmax	Positive local z-coordinate of the top fiber of a beam cross section.

If the element is a rigid translational or rotational link (eltyp = 68 or 69), the material properties are defined as follows.

```
MID   x   y   z   dum   dum   dum   dum   dum   dum   dum   dum   dum   dum   dum   dum   dum   dum
dum
```

MID	The material id number from element attribute table.
x	Any nonzero value implies a rigid motion in this direction (float).
y	Any nonzero value implies a rigid motion in this direction (float).
z	Any nonzero value implies a rigid motion in this direction (float).
dum	Any dummy value (float).

If the element is a joint (eltyp = 120, 122, 123, 124, or 125) or rigid element (eltyp = 65, 66, 67, 68, 69, 70, 71, 73, or 74), the method for enforcing the associated constraints (see CONSTRAINTS) can be specified as follows.

```
MID   CONSTRAINT_METHOD
```

MID	The material id number from element attribute table.
CONSTRAINT_METHOD	Specified method for enforcing the constraints associated with the joint or rigid elements sharing the material id number MID (characters). The default method is set in CONSTRAINTS and used whenever this entry is omitted.
multipliers	The Lagrange multiplier method.
elimination	The elimination method.
penalty [beta]	The penalty method. The parameter beta should be a large positive number, typically of the order of 10^8 (no default value is provided).

If the element is an actuated joint (eltyp = 126 or 127), the material properties may be defined as follows.

```
MID   CONSTRAINT_METHOD   AMPLITUDE   CIRCULAR_FREQUENCY
```

MID	The material id number from element attribute table.
CONSTRAINT_METHOD	Specified method for enforcing the constraints associated with the actuated joint elements sharing the material id number MID (characters). The default method is set in CONSTRAINTS and used whenever this entry is omitted.
multipliers	The Lagrange multiplier method.

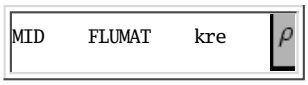
elimination	The elimination method.
penalty [beta]	The penalty method. The parameter beta should be a large positive number, typically of the order of 10^8 (no default value is provided).
AMPLITUDE	Amplitude of the harmonic actuation (float).
CIRCULAR_FREQUENCY	Circular frequency of the harmonic actuation in radians/sec (float).

If the element is a thermal element (eltyp = 3, 10, 46–51, 53, 56–58, 81–86, 4646), the material properties are defined as follows.

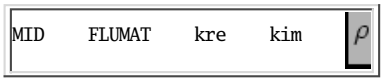
MID	THERMMAT	A	ρ	c_p	h/ϵ	σ	k	t	P	Tr

MID	The material id number from element attribute table.
THERMMAT	keyword that specifies that the following data entries are associated with thermal elements.
A	Cross sectional area for lineal thermal elements (float).
ρ	Mass density per unit volume (float).
c_p	Specific heat coefficient (float).
h/ϵ	Heat convection coefficient for a heat convection or bulk fluid thermal element, or transfer factor for a heat radiation element (float). When radiation is exchanged between two bodies Ω_1 (identified here by this value of MID) and Ω_2 (any other body), the transfer factor $\mathcal{F}_{1-2}$ depends on the emittances of both bodies as well as the geometrical view. In the special case of a gray object in a large environment — that is, when Ω_1 represents a smaller body and Ω_2 a larger isothermal environment (for example, the atmosphere at some temperature) — $\mathcal{F}_{1-2}$ becomes the emissivity ϵ_1 of the first body. In this case, $\mathcal{F}_{1-2} = \epsilon_1 = 1$ if Ω_1 is furthermore a black body.
σ	Stefan's constant, also known as the Stefan-Boltzmann constant (in SI units, $\sigma = 5.670400 \times 10^{-8}$ Js $^{-1}$ m $^{-2}$ K $^{-4}$) (float).
k	Heat conduction coefficient (float).
t	Thickness of a (two-dimensional) thermal element (float).
P	Perimeter/circumference area for thermal elements (float), or depth of the boundary where convection (radiation) occurs when element type 47 (56) is used to model non-lateral boundary convection (radiation) (float).
Tr	For a heat radiation element, reference temperature of the enclosure receiving the radiation (float).

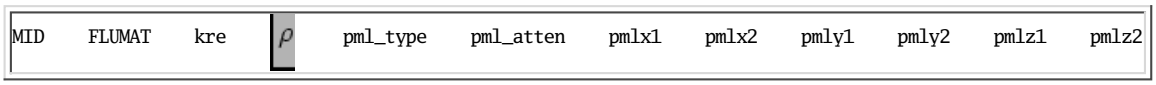
If the element is an acoustic element (eltyp = 31–45, 63, 90, 93–96, 98–108), the material properties may be defined as follows.



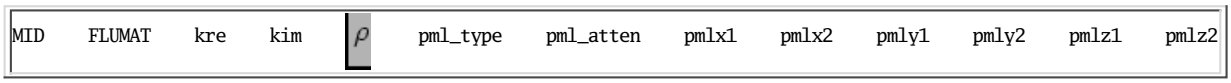
or



or

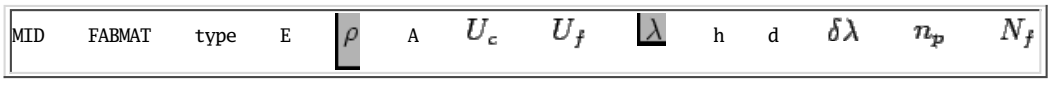


or



MID	The material id number from element attribute table.
FLUMAT	Keyword that specifies that the following data entries are associated with acoustic elements.
kre	Real part of the effective wave number in the fluid (float).
kim	Imaginary part of the effective wave number in the fluid (float).
ρ	Mass density per unit volume (float).
pml_type	Type of PML (Perfectly Matched Layer) (integer). pml_type = 1 designates a box PML. In this case the PML elements are assumed to be in the region defined by pmlx1 < x < pmlx2, -pmlx2 < x < -pmlx1, pmly1 < y < pmly2, -pmly2 < y < -pmly1, pmlz1 < z < pmlz2, and -pmlz2 < z < -pmlz1. pml_type = 2 designates a spherical PML. In this case the PML elements are assumed to be in the region defined by pmlx1 < r < pmlx2, where r = sqrt(x*x+y*y+z*z). pml_type = 3 designates a cylindrical PML. In this case the PML elements are assumed to be in the region defined by pmlx1 < r < pmlx2, where r = sqrt(x*x+y*y), pmlz1 < z < pmlz2, and -pmlz2 < z < -pmlz1.
pml_atten	PML attenuation parameter (float).

If the element is a fabric truss element (eltyp = 111), the material properties are defined as follows (see Powell, D.A. and Zohdi, T.I. Attachment mode performance of network-modeled ballistic fabric shielding. *Composites: Part B* 2009; 40: 451-460).



MID	The material id number from element attribute table (integer).
FABMAT	Keyword specifying that the following data entries are associated with fabric elements (characters).
type	Type of fabric material (integer). If type = 1, the fabric properties are automatically determined by **FEM** using a micro-scale computation. If type = 2, they are determined using Gaussian distribution.
E	If type = 1, this is Young's modulus for the fibrils that make up the yarn (float). If type = 2, it is the mean value of Young's modulus for the yarn.

ρ	Mass density per unit volume of the fibril/yarn (float).
A	Cross sectional area of the entire yarn element (float).
U_c	If type = 1, this is the breaking stretch (l/l_0) of the fibrils (float). If type = 2, it is the stretch level at which the yarn begins to damage.
U_f	Stretch level at which the entire yarn has failed (i.e. the stress response is approximately zero) (float).
λ	If type = 1, this is the initial guess for the parameter controlling the damage rate of the yarn; its actual value is positive and determined by Newton's method (float). If type = 2, it is the slope of the assumed linear variation of the damage parameter with Young's modulus.
h	If type = 1, this is the length of an unstretched yarn (initial length of the truss element) (float). If type = 2, it is the value of the damage parameter for a zero Young's modulus.
d	If type = 1, this is the standard deviation for the inclination distance of the fibrils (misalignment of the fibrils) (float). If type = 2, it is the standard deviation associated with the assumed Gaussian distribution of Young's modulus.
$\delta\lambda$	If type = 1, this parameter is set to 0 and ignored (float). If type = 2, it is the standard deviation associated with the assumed Gaussian distribution of the damage parameter.
n_p	If type = 1, this is the maximum number of Newton iterations for fitting the damage parameter λ to micro-scale data (integer). If type = 2, this parameter is ignored.
N_f	If type = 1, this is the number of fibers in a typical yarn (integer). If type = 2, this parameter is ignored.

Next: LOAD, Previous: MATERIAL

63 LINEAR MULTIPOINT CONSTRAINTS

Command Statement:	**LMPC**

The LMPC command statement is used to specify a set of linear multipoint constraint equations of the form

$$\sum_{j=1}^{j=n} c_{ij} u_j = r_i, \quad i = 1, 2, \ldots$$

in the structural model. There is no limitation on the number of multipoint constraints, or number of degrees of freedom related by the same constraint equation.

The format of this command statement is as follows.

Note 1: The Lagrange multiplier method for enforcing the constraints associated with this command is supported only by the FETI-DP family of solvers, the GMRES solver, and the SPOOLES and MUMPS direct sparse solvers with pivoting enabled, and in all but explicit dynamic analyses (see STATICS).

Note 2: For time-dependent problems, the specified initial conditions must verify the specified linear multipoint constraints.

```
LMPC
```

```
CONSTRAINT# RHS  CONSTRAINT_METHOD
NODE#        DOF# COEFF
.
.
NODE#        DOF# COEFF
```

CONSTRAINT#	This corresponds to the constraint equation number *i* (integer).
RHS	This is the right-hand side r_i of the *i*-th constraint equation (float).
CONSTRAINT_METHOD	This is the method for enforcing the constraint (characters). The default method is set in CONSTRAINTS and used whenever this entry is omitted.
multipliers	The Lagrange multiplier method.
elimination	The elimination method.
penalty [beta]	The penalty method. The parameter beta should be a large positive number, typically of the order of 10^8 (no default value is provided).
NODE#	This is the number of the node contributing the coefficient c_{ij} of the *i*-th constraint equation (integer).
DOF#	This is the local number of the degree of freedom at the node specified above contributing the coefficient c_{ij} of the *i*-th constraint equation (integer).
COEFF	This is the coefficient c_{ij} of the *i*-th constraint equation (float).

Next: LUMPED, Previous: LMPC

64 LOAD

Command Statement:	**LOAD**

The LOAD command statement is used to inform FEM where the user defined subroutines for user defined forces and/or displacements and/or control are located. An example input file using the LOAD command can be found in FEM.d/fem_examples/Control.d

```
LOAD pathandfilename
```

pathandfilename	Specifies **between quotes " "** the path and filename of the LOAD file. The extension of this file must be ".so".

Next: MASS, Previous: LOAD

65 LUMPED

Command Statement:	**LUMPED**

By default, **FEM** computes all element mass matrices and gravity (see GRAVITY) loads by a consistent approach, **except for explicit dynamic computations** (see DYNAMICS), in which case **FEM** always uses a lumped approach for this purpose. If a consistent mass matrix is not available for a particular element (see TOPOLOGY), then **FEM** uses in all cases a lumped mass matrix and gravity load for that element.

Alternatively, this command can be used to instruct **FEM** to compute all element mass matrices and gravity loads by a lumping method.

```
LUMPED
```

Next: MATLAW, Previous: LUMPED

66 MASS EVALUATION

Command Statement:	**MASS**

The `MASS` command statement is used to signal that the user would like that the total mass of the structural system be computed. The result is output on the screen by FEM. In addition to the total mass of the structure, the center of gravity (cg), the center of volume, and the closest node to the cg are also computed and printed on the screen.

Next: MATUSAGE, Previous: MASS

67 MATERIAL LAW

Command Statement:	**MATLAW**

The `MATLAW` command statement can be used to define one or several general material laws, whether they are linear or nonlinear. Typically, the user will write his/her own software describing a material law and package it in a library with a ".so" extension. The `MATLAW` command is then used to label this material law and pass to its library the parameters it expects. The user should also grab the special makefile and include files for this command which are located within the FEM.d/Matlaw.d directory.

The `MATLAW` command can also be used to specify two material laws that have been predefined and implemented in **FEM**. These are the "linear" material law, and the "bilinplast" material law (bilinear plastic material).

The input format of this command is given below.

```
MATLAW
```

READ	material_name	pathandfilename				
MATERIAL_LABEL	MATERIAL_NAME	PARA#1	...	PARA#N		
MATERIAL_LABEL	linear	ρ	E	ν		
MATERIAL_LABEL	bilinearplastic	ρ	E_1	ν	E_2	σ_{yield}
MATERIAL_LABEL	neohookean	ρ	E	ν		
MATERIAL_LABEL	hypoelastic	PARA#1	...	PARA#N		
MATERIAL_LABEL	j2plasticity	PARA#1	...	PARA#N		

READ	Sub-command keyword used to introduce a material name (characters) and associate it with a library to be found at “pathandfilename.so” (characters) and linked with **FEM**. The path and filename of the material law library file must be specified between quotes (“ ”). The extension of this file must be .so.
MATERIAL_LABEL	The material law id number (integer).
MATERIAL_NAME	Name of the material law (characters).
PARA#1	First parameter expected by the library pathandfilename.so.
PARA#N	Last parameter expected by the library pathandfilename.so.
linear	Name of the predefined usual linear material law.
ρ	Mass density per unit volume (real).
E	Young's Modulus (real).
ν	Poisson's ratio (real).
E_1	Young's Modulus for the bilinear plastic material law (real).
E_2	Second slope for the bilinear plastic material law (real).
σ_{yield}	Yield stress for the bilinear plastic material law (real).
PARA#1	Young's modulus (real).
PARA#2	Poisson's ratio (real).
PARA#3	Mass density (real).
PARA#4	Yield stress (j2plasticity) (real).
PARA#5	Hardening modulus (j2plasticity) (real).
PARA#19	Shell shear correction factor (real).
PARA#20	Shell thickness (real).

Next: MODE, Previous: MATLAW

68 MATERIAL LAW USAGE

Command Statement:	**MATUSAGE**

The MATUSAGE command statement can be used to assign a material law specified in the MATLAW command to one or several elements. The input format of this command is

```
MATUSAGE
```

```
ELEMENT#          MATLAW_LABEL#
```

or

```
STARTING_ELEMENT#          ENDING_ELEMENT#          MATLAW_LABEL#
```

ELEMENT#	Element number whose material law number is to be specified (integer).
MATLAW_LABEL	Material law identification number (integer).
MATERIAL_LABEL	First element of a sequence of elements that have the same MATLAW_LABEL (integer).
MATERIAL_NAME	Last element of a sequence of elements that have the same MATLAW_LABEL (integer).

Next: NODALCONTACT, Previous: MATUSAGE

69 SAVING EIGENMODES OR PROJECTING ONTO EIGENVECTORS

Command Statement:	**MODE**

The MODE command is used to signal that: (a) when the EIGEN command is used, the eigen solutions are to be saved in a binay file named EIGENMODES (or binary files EIGENMODES# when using **FEM** in distributed memory mode with one file per MPI process) and located in the execution path of **FEM**, (b) when the DYNAMICS command is used, the binay file named EIGENMODES (or binary files EIGENMODES# when using **FEM** in distributed memory mode with one file per MPI process) located in the execution path of **FEM** is read, and the projections of the transient solution onto the eigen modes found in that file are computed (and outputted if requested in the OUTPUT command), and (c) when the IMPEDANCE command is used with FREQSWEEP, PADEPOLES and ALG = PadeLanczos, the binay file named EIGENMODES (or binary files EIGENMODES# when using **FEM** in distributed memory mode with one file per MPI process) located in the execution path of **FEM** is read, and the eigen modes stored in this (or these) file(s) are exploited to identify within a range of interest the eigenvalues missed by a previous eigen computation and their corresponding eigenvectors.

Note 1: The EIGENMODES file is written using the “internal” numbering of **FEM** for the degrees of freedom. This internal

numbering depends on the equation solver and renumbering scheme that were specified under the STATICS and RENUMBERING commands, respectively, when the eigen solutions were computed. Hence, when reusing the EIGENMODES file for computing the projections of the transient solution of a problem onto the vectors contained in this file, the same equation solver and renumbering scheme must be specified under the STATICS and RENUMBERING commands, respectively.

Next: NODES, Previous: MODE

70 NODAL CONTACT

Command Statement:	**NODALCONTACT**

The NODALCONTACT command can be used to specify node-to-node contact of the form

$$(u_2 - u_1) \cdot n_{12} = 0$$

or

$$(u_2 - u_1) \cdot n_{12} \geq gap$$

where *u* denotes the displacement field, n_{12} denotes the normal to the contact surface oritented from node 1 to node 2, and gap denotes the initial gap. The input format of this command can be as follows.

Note 1: For mode_v = 1 and mode_v = 3, the enforcement of the nodal contact constraints by the Lagrange multiplier method is supported only by the FETI-DP family of solvers, and for static and implicit dynamic analyses only.

Note 2: For mode_v = 1 and mode_v = 3, the enforcement of the nodal contact constraints by the elimination method is not supported.

Note 3: For mode_v = 0 and mode_v = 2, the enforcement of the nodal tied contact constraints by the Lagrange multiplier method is supported only by the FETI-DP family of solvers, the GMRES solver, and the SPOOLES and MUMPS direct sparse solvers with pivoting enabled, and in all but explicit dynamic analyses.

```
NODALCONTACT
```

or

```
NODALCONTACT        MODE        default_mode_v
```

followed by one or more lines of the form

```
NODE_1        NODE_2        N_X        N_Y        N_Z
```

or

```
NODE_1      NODE_2      N_X      N_Y      N_Z      GAP      gap_v
```

```
NODE_1      NODE_2      N_X      N_Y      N_Z      MODE      mode_v
```

```
NODE_1      NODE_2      N_X      N_Y      N_Z      MODE      mode_v      GAP      gap_v
```

```
NODE_1      NODE_2      N_X      N_Y      N_Z      MODE      mode_v      GAP      gap_v
CONSTRAINT_METHOD
```

NODE_1	Id number of the first node (integer).
NODE_2	Id number of the second node (integer).
N_X	x-component of the normal to the contact surface oriented from NODE_1 to NODE_2 (float).
N_Y	y-component of the normal to the contact surface oriented from NODE_1 to NODE_2 (float).
N_Z	z-component of the normal to the contact surface oriented from NODE_1 to NODE_2 (float).
GAP	Keyword to be spelled out (characters).
gap_v	Initial gap between NODE_1 and NODE_2. A negative gap means an initial penetration (float).
MODE	Keyword to be spelled out (characters).
mode_v	Specifying mode_v = 0 designates a tie-of-the-normal-component-of-the-displacement constraint. Specifying mode_v = 1 designates a normal contact constraint. Specifying mode_v = 2 designates a tie-of-the-normal-and-tangential-components-of-displacment constraint. Specifying mode_v = 3 desinates both normal contact and tie-of-the-tangential-components-of-displacement constraints (integer).
default_mode_v	Default mode_v for all constraints. If not specified, the default mode is set to 1, i.e. normal contact (integer).
CONSTRAINT_METHOD	This is the method for enforcing the associated constraints (characters). The default method is set in CONSTRAINTS and used whenever this entry is omitted.
multipliers	The Lagrange multiplier method.
elimination	The elimination method.
penalty [beta]	The penalty method. The parameter beta should be a large positive number, typically of the order of 10^8 (no default value is provided).

Next: NONINPC, Previous: NODALCONTACT

71 NODES *S*

Command Statement:	**NODES**

The NODES command statement is used to signal that the following data lines correspond to the coordinates of each node. The input data format is given below. Note that there should be as many lines as the number of nodes in the mesh. If the Y-ordinate and Z-ordinate are not given for a node, FEM assumes they are zero. Similarly, if the Z-ordinate is not given, FEM assumes it is zero.

```
NODES
```

```
NODE#        X-ORDINATE        Y-ORDINATE        Z-ORDINATE
```

NODE#	Node number whose coordinates are to be specified (integer).
X-ORDINATE	X-ordinate of the given node number (float).
Y-ORDINATE	Y-ordinate of the given node number (float).
Z-ORDINATE	Z-ordinate of the given node number (float).

Next: NONLINEAR, Previous: NODES

72 NON INTRUSIVE POLYNOMIAL CHAOS

Command Statement:	**NONINPC**

The NONINPC command signals to the FEM code to perform a non intrusive uncertainty quantification analysis using the Polynomial Chaos (PC) representation. It requires that the input file also contains the GROUPS, RANDOM, and STATICS commands as well as their respective data. Essentially, FEM computes in this case the mean and standard deviation of a computed primal field v as follows

$$\bar{v} = \int_{-\infty}^{+\infty} v p(v) dv = v_0 = \frac{\sum_{i=1}^{N} v^{(i)}}{N}$$

$$\sigma_v = \sqrt{\int_{-\infty}^{+\infty} (v - \bar{v})^2 p(v) dv} = \sqrt{\sum_{i=1}^{i=P-1} \left(\int_{-\infty}^{+\infty} \psi_i^2 p(\psi_i(\{\xi_j\})) d\psi_i \right) v_i^2}$$

$$= \sqrt{\sum_{i=1}^{i=P-1} \left(\int_{-\infty}^{+\infty} \psi_i^2 p(\psi_i(\{\xi_j\})) d\psi_i \right) \left(\frac{\sum_{j=1}^{N} v^{(j)} \psi_i^{(j)}}{N \int_{-\infty}^{+\infty} \psi_i^2 p(\psi_i) d\psi_i} \right)^2}$$

where v_0 denotes the constant of the PC expansion of the non deterministic v field ("the output"), N denotes the number of realizations, $v^{(i)}$ denotes the deterministic solution of the field v associated with the i-th realization of the random system, p

denotes the probability density function, ξ_j denotes the j-th "basic" random variable which is assumed to be Gaussian, v_i denotes the deterministic coefficient of the i-th PC shape function ψ_i, P denotes the total number of PC shape functions ψ_i representing the non deterministic output, and $\psi_i^{(j)}$ denotes the evaluation of ψ_i at the j-th realization of the random system.

Note 1: Currently, this command is limited to the post-processing of the results of a linear, static, structural analysis.

The input format of this command is given below.

```
NONINPC
```

```
DEG_OUT          NUM_RLZ
```

DEG_OUT Degree of the PC representation of the output which, together with the data of the RANDOM command, determines P.

NUM_RLZ Number of realizations N of the random system (which implies the number of deterministic solutions) (integer).

Next: OUTPUT, Previous: NONINPC

73 NONLINEAR

Command Statement:	**NONLINEAR**

The NONLINEAR command signals to the **FEM** code that the current statics or dynamics analysis is to be nonlinear. For structural problems, the nonlinear analysis is by default a geometrically nonlinear analysis where the tangent stiffness matrix and/or internal force is built using the Corotational Finite Element Method for some elements and the Lagrangian Method for others. For structural problems, materially and geometrically + materially nonlinear analyses are also possible but require, in addition to this command, the selection under the TOPOLOGY command of element types that support such analyses. Also for structural problems, material nonlinearities can be specified and described using the MATLAW command, among others. For nonlinear static and implicit dynamic analyses, only Newton-like methods are available.

Note 1: Currently, geometric stiffnesses are implemented only for the bar, beam, 3-noded shell and composite shell, 4-noded shell and composite shell, 4-noded tetra, 8-noded brick, spring, and rigid elements (via their conversion to linear multipoint constraints).

Note 2: Currently, the list of materially nonlinear elements includes only an 8-noded brick element.

Note 3: Currently, the list of geometrically + materially nonlinear elements includes only an 8-noded brick element.

The input format is given below.

```
NONLINEAR
```

```
METHOD (keyword)
PARAMETERS (keyword and/or value)
```

METHOD

arclength	Selects the arclength method for solving the nonlinear equations associated with a nonlinear static or an implicit dynamic computation. If this option is not selected, then the default algorithm for solving the nonlinear quations is a standard Newton-Raphson iteration method (string).

PARAMETERS

maxitr	For a nonlinear static or an implicit dynamic computation, specifies the maximum number of Newton-like iterations to be performed (per time-step) (integer). The default value is 100.
nltol	For a nonlinear static or an implicit dynamic computation, specifies the tolerance for convergence of the Newton-like method (float). The default value is 1e-06.
dlambda float float	Specifies the loading parameters to use in a nonlinear static or implicit dynamic analysis. The first float entered is the load fraction increment, and the second float is the total load factor. For example, “dlambda 0.25 1.0” would mean to run 4 load steps each applying an incremental load of 0.25*(Applied Force) per load step. On the other hand, “dlambda 0.25 2.0” means 8 load steps each applying an incremental load of 0.25*(Applied Force) per load step. This can help the convergence of Newton-Raphson.
fitalg	For structural problems, specifies the fitting algorithm for the corotational formulation. 8or `fitalg = 1` (default value) the tangent stiffness matrix 8s not consistent but the corotational framework is most robust. For `fitalg = 2`, the tangent stiffness matrix is consistent (integer).
nlmat	For structural problems, specifies a materially or geometrically + materially nonlinear analysis. In that case, appropriate element types must be specified in the TOPOLOGY command. If this keyword is absent, the NONLINEAR command requests a geometrically nonlinear analysis (string).

Next: OUTPUT6, Previous: NONLINEAR

74 OUTPUT OF RESULTS (OUTPUT completely spelled out)

Command Statement:	**OUTPUT**

The OUTPUT command statement is used to signal that the following lines of data correspond to specifying which results are to be outputted. For vector results such as displacement fields, this command forces **FEM** to output only the three translational components.

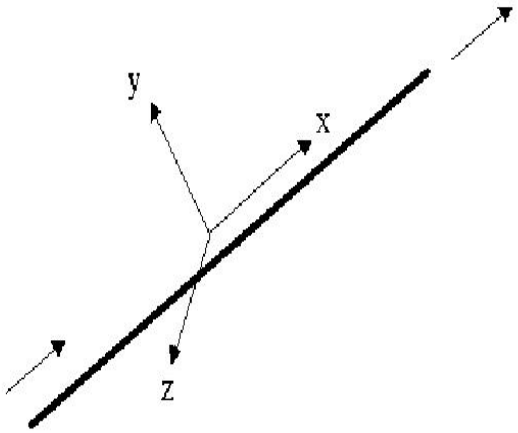
The shared memory version of **FEM** generates ASCII output files. Most of these are in a format that is suitable for postprocessing by the **XPost** sofware, but some are in a format that is suitable for postprocessing by **gnuplot**. The distributed memory version of **FEM**

generates binary output files except for the case of selective nodal output which it generates in ASCII format. The **SOWER** software can be used to convert the binary output files into ASCII output files, in a format suitable for postprocessing by **XPost**. All "nodal" output files are generated in the **gnuplot** format. Examples using the various OUTPUT commands can be found in FEM.d/fem_examples/Output.d/.

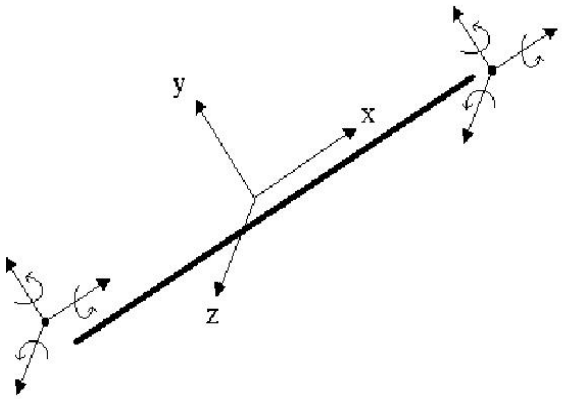
Note 1: In addition to the time-instances implied by the specified value of the parameter INCREMENT, any result specified for output is also outputted at the beginning of a simulation if INCREMENT is non zero, and at the of a simulation if **FEM** exits gracefully.

Note 2: The strain and stress outputs are not available for the bar and beam elements, except for STRAINXX and STRESSXX

Note 3: The internal forces and moments and STRAINXX and STRESSXX for the bar and beam elements are outputted in the local frames using the sign conventions graphically depicted below.



Sign conventions for the output of the internal forces in a bar element



Sign conventions for the output of the internal forces in a beam element

Note 3: For nonlinear structural analysis, the stresses and strains for the corotational elements are outputted on the deformed configuration and in the local frames.

Note 4: For nonlinear structural analysis, the stresses and strains for the total Lagrangian elements are outputted on the undeformed configuration and in the global frame.

Note 5: For nonlinear structural analysis, the internal forces and moments and `STRAINXX` and `STRESSXX` for the bar and beam corotational elements are outputted in the local frames associated with the deformed configuration.

Note 6: When multiple outputs are expected, they are printed one set after the other in the same output file.

Note 7: To visualize the directions of principal strains and stresses, the auxiliary code PDIREC should be used to generate the node set and element set used by **XPost** for this purpose. When runnig PDIREC, the user is asked to enter the name of the mesh **XPost** file (for example, the one obtained from executing **FEM** with the "-t" command), a **FEM** result file, the type of that result for labeling purposes (eg. `SP1DIREC`), and a characteristic length to be used for scaling the length of the elements representing the directions to be visualized. PDIREC can also be used for visualizing any other vector output (eg. `GDISPLAC` or `GVELOCIT`).

Note 8: The vocabulary used for the request of stresses and strains that are associated with the elements listed below differs from the standard continuum mechanics notation. See below for specific notations.

The input format of this command can be as follows.

```
OUTPUT
```

```
RESULT [FORMAT] PATHANDFILENAME INCREMENT [NODE_NUM BER] [OPTION] [...]
[OPTION]
```

or

```
RESULT [FORMAT] PATHANDFILENAME INCREMENT [N NODE_NUMBER] [OPTION] [...]
[OPTION]
```

or

```
RESULT [FORMAT] PATHANDFILENAME INCREMENT [NG GROUP_NUMBER] [OPTION] [...]
[OPTION]
```

RESULT	The following results can be outputted:
GDISPLAC	Generalized nodal displacements.
DISPLACX	Nodal displacements in the x direction.
DISPLACY	Nodal displacements in the y direction.
DISPLACZ	Nodal displacements in the z direction.
ROTATIOX	Nodal rotations in the x direction.
ROTATIOY	Nodal rotations in the y direction.
ROTATIOZ	Nodal rotations in the z direction.
DISPLMOD	Euclidean norms of the nodal displacements.
ROTATMOD	Euclidean norms of the nodal rotations.
GDISPMOD	Euclidean norms of the generalized nodal displacements.
GVELOCIT	Nodal velocities.
GACCELER	Nodal accelerations.
GTEMPERA	Nodal temperatures.
GTEMPVEL	Nodal first time-derivatives of the temperature.
GEIGENPA	Eigenvalues and Eigenvectors. If singular modes are found during an eigen analysis, they are saved in the file FNAME associated with GEIGENPA *before* the non-singular eigen results.
GEIGSLSH	Sloshing Eigenvalues and Eigenvectors (displacement potential). If a singular mode is found during the analysis, it is saved in the file FNAME associated with GEIGSLSH *before* the non-singular eigen results.
SLSHDISP	True fluid nodal displacements associated with the computed sloshing modes.
STRAINXX	Strains in the global xyz-axes.
STRAINYY	Strains in the global xyz-axes.
STRAINZZ	Strains in the global xyz-axes.
STRAINXY	Strains in the global xyz-axes.
STRAINYZ	Strains in the global xyz-axes.
STRAINZX	Strains in the global xyz-axes.
STRAINVM	von Mises strain.
EFFPSTRN	Effective plastic strain. **Currently, this result is available only for the Belytschko-Tsay element (type = 16)**
STRAINP1	First principal strains.
STRAINP2	Second principal strains.

STRAINP3	Third principal strains.
EP1DIREC	Directions of first principal strains.
EP2DIREC	Directions of second principal strains.
EP3DIREC	Directions of third principal strains.
STRESSXX	(Second-Piola) stresses in the global xyz-axes.
STRESSYY	(Second-Piola) stresses in the global xyz-axes.
STRESSZZ	(Second Piola) stresses in the global xyz-axes.
STRESSXY	(Second Piola) stresses in the global xyz-axes.
STRESSYZ	(Second Piola) stresses in the global xyz-axes.
STRESSZX	(Second Piola) stresses in the global xyz-axes.
STRESSVM	(Second Piola von Mises stress in the element.
STRESSP1	(Second Piola) first principal stresses.
STRESSP2	(Second Piola) second principal stresses.
STRESSP3	(Second Piola) third principal stresses.
SP1DIREC	Directions of first principal stresses.
SP2DIREC	Directions of second principal stresses.
SP3DIREC	Directions of third principal stresses.
HEATFLXX	Temperature flux along global x.
HEATFLXY	Temperature flux along global y.
HEATFLXZ	Temperature flux along global z.
GRDTEMPX	Temperature gradient along global x.
GRDTEMPY	Temperature gradient along global y.
GRDTEMPZ	Temperature gradient along global z.
INXFORCE	Internal forces along local x-axis.
INYFORCE	Internal forces along local y-axis.
INZFORCE	Internal forces along local z-axis.
AXMOMENT	Applied (internal) moments along local x-axis.
AYMOMENT	Applied (internal) moments along local y-axis.
AZMOMENT	Applied (internal) moments along local z-axis.
ENERGIES	External, Aeroelastic, Elastic, Kinetic, Damping, and Numerical Production energies. The Numerical Production energy corresponds to the energy conservation error for a conservative system. When positive, it suggests a numerically unstable scheme. When negative, it suggests the presence of numerical damping in the scheme. Using the standard notation and that of the DYNAMICS section, these energies are computed as follows:

External :

$$\sum_{k=0}^{n-1}[(1-\gamma)F^{ext^k}+\gamma F^{ext^{k+1}}]^T(u^{k+1}-u^k)$$

Aeroelastic :

$$\sum_{k=0}^{n-1}[(1-\gamma)F^{aer^k}+\gamma F^{aer^{k+1}}]^T(u^{k+1}-u^k)$$

Elastic :	$u^{n^T} K u^n$
Kinetic :	$\dot{u}^{n^T} M \dot{u}^n$
Damping :	$\sum_{k=0}^{n-1} [(1-\gamma) C \dot{u}^k + \gamma C \dot{u}^{k+1}]^T (u^{k+1} - u^k)$
Numerical Production :	Kinetic - Previous_Kinetic + Elastic - Previous_Elastic + Increment_Damping_Between_Previous_and_Current - Increment_Exterior_Between_Previous_and_Current

AEROFORX Aeroelastic nodal forces along global x-axis at t^{n+1}, after application of the boundary conditions (see DYNAMICS).

AEROFORY Aeroelastic nodal forces along global y-axis at t^{n+1}, after application of the boundary conditions (see DYNAMICS).

AEROFORZ Aeroelastic nodal forces along global z-axis at t^{n+1}, after application of the boundary conditions (see DYNAMICS).

AEROMOMX Aeroelastic nodal moments along global x-axis at t^{n+1}, after application of the boundary conditions (see DYNAMICS).

AEROMOMY Aeroelastic nodal moments along global y-axis at t^{n+1}, after application of the boundary conditions (see DYNAMICS).

AEROMOMZ Aeroelastic nodal moments along global z-axis at t^{n+1}, after application of the boundary conditions (see DYNAMICS).

RAEROFOR Aeroelastic resultant forces along global x, y, and z axes (integrated over the wet surface of the structure) at t^{n+1}, before application of the boundary conditions (see DYNAMICS).

RAEROTFL Fluid heat fluxes transferred to the structure at t^{n+1}.

GHELMHOL Real or complex acoustic pressure scattered field computed by an acoustic analysis in the frequency domain. The real part is always output first, then, if it is non uniformly zero, the imaginary part is output as a separate frame but in the same output file.

GACOUPRE Nodal values of the acoustic pressure computed by an acoustic analysis in the time domain.

GACOUVEL Nodal values of the first time-derivative of the acoustic pressure computed by an acoustic analysis in the time domain.

GACOUACC Nodal values of the second time-derivative of the acoustic pressure computed by an acoustic analysis in the time domain.

KIRCHHOFF Acoustic pressure field at locations specified using the KIRLOC command and evaluated using the Kirchhoff integral. This result can be outputted only if the surface of the scatterer is defined using the command HSCB. The corresponding output file contains on each line five words of the form: X-ORDINATE_I Y-ORDINATE_I Z-ORDINATE_I REAL(U_I) IMAG(U_I). X, Y, and Z-ORDINATE_I are the X, Y, and Z coordinates of the I-th point specified using the command KIRLOC. REAL(U_I) and IMAG(U_I) are the real and imaginary parts of the acoustic pressure field computed at the I-th specified point using the Kirchhoff integral method.

FARFIELD Farfield pattern of the acoustic pressure field. This field can be output only if the HSCB command is also specified. In this case, the output file contains on each line four words of the form "α, β, real-part(ffp), imaginary-part(ffp)". The first two quantitites are the spherical angles (expressed in

	radians) that determine the direction $d = [\cos\beta * \cos\alpha,\ \cos\beta * \sin\alpha,\ \sin\beta]$. The number of output lines depends on the parameter INCREMENT (see below).
GENCOORD	Projections using a mass-based metric of a transient solution on the eigen modes saved by the MODE command in the file EIGENMODES. This field can be output only if the MODE command is also specified.
MODERROR	Relative truncation error using the two-norm of the representation of a transient solution by its projection on the eigen modes saved by the MODE command in the file EIGENMODES. This field can be output only if the MODE command is also specified.
MODALDSP	Modal coefficients of the global displacements, when dynamics computations are done in a modal basis.
MODALEXF	Modal coefficients of the external forces, when dynamics computations are done in a modal basis.
CONPRESS	Contact pressure forces (Lagrange multipliers in the normal directions).
CONFACE	Denotes the status of interactions at a node. A value of 0.5 indicates the node is not in contact. A value of 1, 2, or 3 denotes the number of interactions at that node.
NORMAL_FORCE_MAG	Normal force magnitude at a node. If multiple constraints exist at a node, the value is for the last constraint.
NORMAL_TRACTION_MAG	Normal traction magnitude at a node. If multiple constraints exist at a node, the value is for the last constraint.
TANGENTIAL_FORCE_MAG	Tangential force magnitude at a node. If multiple constraints exist at a node, the value is for the last constraint.
TANGENTIAL_TRACTION_MAG	Tangential traction magnitude at a node. If multiple constraints exist at a node, the value is for the last constraint.
CDIRNORX	x component of the normal direction for the constraint at a node. If multiple constraints exist at a node, the value is for the last constraint.
CDIRNORY	y component of the normal direction for the constraint at a node. If multiple constraints exist at a node, the value is for the last constraint.
CDIRNORZ	z component of the normal direction for the constraint at a node. If multiple constraints exist at a node, the value is for the last constraint.
CDIRTANX	x component of the tangential direction for the constraint at a node. If multiple constraints exist at a node, the value is for the last constraint.
CDIRTANY	y component of the tangential direction for the constraint at a node. If multiple constraints exist at a node, the value is for the last constraint.
CDIRTANZ	z component of the tangential direction for the constraint at a node. If multiple constraints exist at a node, the value is for the last constraint.
SLIP_MAG	Incremental slip at a node. If multiple constraints exist at a node, the value is for the last constraint.
NODAL_DISSIPATION	Frictional energy dissipated at a node.
CONTACT_AREA	Contact area for a node.
GAP_CUR	Normal gap for the current time. If multiple constraints exist at a node, the value is for the last constraint.
GAP_OLD	Normal gap from the previous time. If multiple constraints exist at a node, the value is for the last constraint.

FORMAT	This optional field can be used to override the default format for outputting a numerical value by one which consists of 2 integer numbers F and P separated by a blank (integers). F specifies the field width — that is, the total number of digits to be printed including the decimal point, the "E" of the exponent and the sign of the exponent, but excluding the sign of the output value. P is the number of digits after the decimal and before the exponent. For example, 8 2 results in printing the output value 1523 as 0.15E+03.
FNAME	User specified filename associated with result (characters).
INCREMENT	Output increment for transient and eigenvalue results (integer). For a FARFIELD output result, this field specifies, in the three-dimensional case, the number of longitudinal directions where the farfield is to be evaluated. In this case, (INCREMENT/2)+1) latitudinal directions are also considered and therefore the farfield is evaluated at ((INCREMENT/2)+1))*INCREMENT points uniformly distributed in spherical coordinates. In two-dimensions, the farfield is evaluated at INCREMENT points that are evenly distributed on a circle. For transient and quasi-static analyses, the final time-instance is always outputted for the requested fields — except for the aeroelastic forces and moments — independently of the value of INCREMENT.
N	Sub-command keyword (character) signaling that the next entry is a node identifying number.
NODE_NUMBER	When this optional field is used, with or without the sub-command keyword N, only the computed results associated with the node NODE_NUMBER are outputted in the same format as otherwise, except that the header of the **XPost** file is ommitted for easier use of a visualization software such as gnuplot (integer).
NG	Sub-command keyword (characters) signaling that the next entry is a group identifying number (see GROUPS).
GROUP_NUMBER	When this optional field is used together with the sub-command keyword NG, only the computed results associated with the nodes belonging to the group of nodes GROUP_NUMBER (see GROUPS) are outputted in the following format which is suitable for a visualization software such as gnuplot (integer). FLOAT-1 NODE_NUMBER-1 X Y Z RESULT-1 RESULT-2 ... RESULT-n NODE_NUMBER-2 X Y Z RESULT-1 RESULT-2 ... RESULT-n ... FLOAT-2 NODE_NUMBER-1 X Y Z RESULT-1 RESULT-2 ... RESULT-n NODE_NUMBER-2 X Y Z RESULT-1 RESULT-2 ... RESULT-n ... where FLOAT-i is an i -th time-instance in the case of a dynamic analysis, frequency-value (number of cycles per second) in the case of a structural dynamics frequency sweep analysis (see IMPEDANCE), or wavenumber in the case of an acoustic (Helmholtz) frequency sweep analysis (see HELMHOLTZ), NODE_NUMBER-j is the j -th node number of group GROUP_NUMBER, X, Y

and Z are the x, y, and z coordinates of NODE_NUMBER-j, and RESULT-1 RESULT-2 ...

RESULT-n are all the computed results pertaining to NODE_NUMBER-j and the chosen instance of RESULT.

OPTION — Any of the following optional fields can be specified after the previous optional and/or non-optional entries documented above:

NDTYPE [NUM_OUT_RLZ] — NDTYPE is an optional keyword that can take one of the following values: MEAN, STDV, or PDF. It should be used only when an intrusive or non intrusive non deterministic analysis is performed. When MEAN is selected and a non intrusive non deterministic analysis is performed, the mean value of the RESULT is outputted using the number of realizations NUM_RLZ specified in the NONINPC command. When STDV is chosen and a non intrusive non deterministic analysis is performed, the standard deviation of the RESULT is outputted using the number of realizations NUM_RLZ specified in the NONINPC command. When PDF is specified, the probability density function is outputted using a number of realizations equal to the integer set in the suboptional field NUM_OUT_RLZ (integer). First, the shape functions are evaluated at NUM_OUT_RLZ realizations of the random system. Then, NUM_OUT_RLZ realizations of the non deterministic RESULT are reconstructed from these evaluations using $v^{(j)} = \sum_{i=0}^{P-1} v_i^{(j)} \psi_i^{(j)}$ in the case of a displacement or $\sigma^{(j)} = \sum_{i=0}^{P-1} \sigma_i^{(j)} \psi_i^{(j)}$ in the case of a stress field σ and written in the output file (note that the case of a von Mises stress is treated slightly differently because it is a nonlinear function of the stress tensor components). A histogram can be constructed from these results.

STR_AVG_OPTION — If the output of a stress or strain field is requested, the user can set this optional field to one of the following values NODALFULL, NODALPARTIAL, ELEMENTAL (characters) to specify a type of stress/strain averaging. If NODALFULL is specified, the stress or strain is averaged at each node using all elements attached to this node. In this case, the contributions of the beam and bar elements are first transformed to the global frame. If NODALPARTIAL is specified, the averaging is performed using only the elements that are neither of a bar or a beam type. If ELEMENTAL is specified, the stress or strain is output without any averaging at each node of each element. Note however that currently, the ELEMENTAL option is not supported when a FETI solver is invoked.

STR_SHELL_OPTION — This optional field is applicable only when the output of a stress/strain field of a shell element is requested. It can take any of the following values UPPER, MEDIAN, LOWER (characters). If UPPER is specified, the stresses are computed on the upper surface of the element. If MEDIAN is specified (default value), they are computed on the median surface of the element. If LOWER is specified, they are computed on the lower surface of the element.

STR_BEAM_OPTION — This optional field is applicable only when the output of STRAINXX or STRESSXX of a beam element is requested. It specifies two real numbers (real). The first (second) real number represents the positive or negative y (z) coordinate in percentage of the maximum positive (top fiber) or negative (bottom fiber) y (z) coordinate of the beam cross section, in the local frame attached to the beam, of the fiber at which ϵ_{xx} or σ_{xx} is requested. The default value for both real numbers is zero.

COMPLEX_OUTPUT_OPTION [NSTEPS] — These optional and suboptional fields are applicable only when the output is complex-valued. COMPLEX_OUTPUT_OPTION can take any of the following values REALIMAG, MODPHASE, ANIM (characters). When set to REALIMAG, the real parts of the complex-valued output are printed first and the imaginary

parts are printed next. When set to MODPHASE, the moduli ρ of the output values are first printed and then the phases θ are printed. When set to ANIM, the values of $\rho\cos(\theta-\phi_1)$ are printed first, followed by the values of $\rho\cos(\theta-\phi_2)$, etc, concluding with the values of $\rho\cos(\theta-\phi_N)$, where ϕ_i is defined as $\frac{2\pi}{N}$ and N is specified in the suboptional field NSTEPS (integer).

For a shell element, the following definitions prevail

STRAINXX	Nodal membrane strains in local xyz-axes.
STRAINYY	Nodal membrane strains in local xyz-axes.
STRAINZZ	Nodal membrane strains in local xyz-axes.
STRAINXY	Nodal bending curvatures in local xyz-axes.
STRAINYZ	Nodal bending curvatures in local xyz-axes.
STRAINZX	Nodal bending curvatures in local xyz-axes.
STRESSXX	Nodal membrane stress resultants in local xyz-axes.
STRESSYY	Nodal membrane stress resultants in local xyz-axes.
STRESSZZ	Nodal membrane stress resultants in local xyz-axes.
STRESSXY	Nodal bending stress resultants in local xyz-axes.
STRESSYZ	Nodal bending stress resultants in local xyz-axes.
STRESSZX	Nodal bending stress resultants in local xyz-axes.

Next: PITA, Previous: OUTPUT

75 OUTPUT OF RESULTS 6 COLUMNS (OUTPUT6 completely spelled out)

Command Statement:	**OUTPUT6**

The OUTPUT6 command has the same options and format of the OUTPUT command. Its outcome differs from that of OUTPUT only when vector results such as generalized displacements fields are to be outputted. In that case, the OUTPUT6 command forces FEM to output all six components of the generalized displacement field in a format slightly different from that of the file created by OUTPUT, in that the node number is included at the beginning of each line.

Note 1: When multiple outputs are expected, they are printed one set after the other in the same output file.

Next: PRELOAD, Previous: OUTPUT6

76 PARALLEL-IN-TIME ALGORITHM (PITA)

Command Statement:	**PITA**

The PITA command statement is used to request the parallelization in time of the time-integration algorithm specified under the DYNAMIC command. This option is supported only when executing the distributed version of the **FEM** executable. The simultaneous presence

of the command statement NONLINEAR in the input file automatically triggers the nonlinear version of the PITA algorithm. By default, the initialization of the time-slices is performed by running the chosen time-integrator on the coarse time-grid. Alternatively, the IDISP6PITA (see IDISP6PITA) and IVEL6PITA (see IVEL6PITA) commands can be used to specify the seed displacements and velocities. The imput format is given below.

Note 1: The current implementation allows only geometric nonlinearities.

Note 2: The foundations of the PITA methodology are described in [Farhat and al.] for the linear case and [Cortial and Farhat] for the nonlinear case.

```
PITA
```

```
J_RATIO        MAX_ITER
PITA_KEYWORDS
```

or

```
J_RATIO        MAX_ITER        MAX_SLICES_ON_CPU
PITA_KEYWORDS
```

J_RATIO	Ratio between the time-steps on the fine and coarse time-grids. Also, number of time-steps in each time-slice (integer).
MAX_ITER	Maximum number of outer PITA iterations (integer).
MAX_SLICES_ON_CPU	Maximum number of time-slices assigned to a processor (integer). If the number of time-slices is larger than the number of available CPUs, the execution of the PITA is organized in phases. In each phase, up to MAX_SLICES_ON_CPU time-slices are assigned to each processor, starting with the first one. The remaining time-slices are then progressively activated in subsequent phases while the active time-slices reach convergence and free new slots. The default value is 1.
PITA_KEYWORDS	None, one or several of the following optional keywords
NOFORCE	Informs **FEM** about the absence of any external force so that a special version of the PITA can be executed for optimal performance (linear case only).
CONSTFORCE	Informs **FEM** that all external forces are time-independent so that a special version of the PITA is executed for optimal performance (linear case only).
CKCOARSE	Instructs **FEM** to propagate the remainder of the correction step on the coarse time-grid (linear case only). This option should be used for testing purposes only. It is not recommended in general.
LOCALBASES	Instructs **FEM** to use the local basis updating scheme instead of the global one (nonlinear case only). This option may improve performance but at the expense of time-accuracy.

Next: PRESSURE, Previous: PITA

77 PRELOAD

Command Statement:	**PRELOAD**

The PRELOAD command statement can be used to specify a preload within one or more truss elements. The preload is an axial force within the bar and affects the element stiffness matrix. It is assumed that the nodal coordinates of the model NODES include the deformations due to the preloading. Preload affects both linear and nonlinear, static, transient, and eigen analyses.

Preload is currently active only for truss elements. Prescribed preloads for other element types are currently ignored. The default preload of all truss elements is zero. When outputting the internal forces of bar elements, note that the axial force will include the preload.

The following two formats are supported and can be mixed.

```
PRELOAD
```

```
ELEMENT#          PRELOAD_VALUE
```

```
STARTING_ELEMENT#          ENDING_ELEMENT#          PRELOAD_VALUE
```

ELEMENT#	Element number where a preload is to be specified (integer).
PRELOAD_VALUE#	Value of the axial preload (float).
STARTING_ELEMENT#	First element of a sequence of elements where the preload has the same constant value (integer).
ENDING_ELEMENT#	Last element of a sequence of elements where the preload has the same constant value (integer).

Next: QSTATICS, Previous: PRELOAD

78 PRESSURE

Command Statement:	**PRESSURE**

The PRESSURE command can be used to specify a piece-wise uniform pressure field on beam and/or shell elements, or surface elements defined in SURFACETOPO. It can be positioned in the input file anywhere after the TOPOLOGY command. For a beam element (type 6 or 7), the pressure is applied in the Y-direction of the **local frame** specified in the EFRAMES command or implied by the third node technique for beam elements (see TOPOLOGY).

For a shell element (type 8, 88, 20, or 2020) or a surface element defined in SURFACETOPO, the pressure field is applied normal to the surface of the element using the usual convention of a positive outward normal — that is, if the three nodes of a shell element are

denoted by *a*, *b*, and *c* and numbered in this order, the normal is computed as $\vec{ab} \wedge \vec{ac}$. Furthermore, a positive value of PRESSURE_VALUE implies a force in the direction of the normal.

In a geometrically nonlinear analysis, the generated pressure forces are of the **follower** type for shell elements and surface elements defined in SURFACETOPO; for beam elements they can either be of the follower type (default) or non-follower type, depending on a compilation switch that must be set for this purpose.

Note 1: FEM always generates pressure loads using a consistent method.

Note 2: The piece-wise uniform pressure field represented by the pressure value PRESSURE_VALUE can be varied in time using the command MFTT.

The following three formats are available for this command and can be mixed.

```
PRESSURE
```

```
ELEMENT#          PRESSURE_VALUE
```

```
STARTING_ELEMENT#          ENDING_ELEMENT#          PRESSURE_VALUE
```

```
SURFACE          SURFACE#          PRESSURE_VALUE
```

ELEMENT#	Element number where a pressure field is to be specified (integer).
PRESSURE_VALUE#	Value of the piece-wise uniform pressure field in that element (float).
STARTING_ELEMENT#	First element of a sequence of elements where the piece-wise uniform pressure field has the same value (integer).
ENDING_ELEMENT#	Last element of a sequence of elements where the piece-wise uniform pressure field has the same value (integer).
SURFACE	Keyword indicating that a surface defined in SURFACETOPO is to be identified next by its integer identification number (characters).
SURFACE#	Integer identification of the surface defined in SURFACETOPO where the uniform pressure value PRESSURE_VALUE is to be applied (integer).

Next: RANDOM, Previous: PRESSURE

79 QUASISTATICS ANALYSIS

Command Statement:	**QSTATICS**

The QSTATICS command statement is used to signal that the subsequent data lines correspond to the following quasistatics *iterative* algorithm for solving *Ax=b*, where *A* can represent a diagonal block of a coupled system and can be singular

$$A\tilde{x} = b^{n+1}$$
$$\alpha^{n+1} = \alpha^n + \beta R^T b^{n+1}$$
$$\bar{x}^{n+1} = \bar{x}^n + \theta(\tilde{x} - \bar{x}^n)$$
$$x^{n+1} = \bar{x}^{n+1} + R\alpha^{n+1}$$

This command is currently implemented only for linear analysis, and is useful only for a multidisciplinary analysis (aerothermal, aeroelastic, aerothermoelastic).

For single discipline analysis, it is at best ($\theta = 1$) a direct solution method.

The second and fourth steps of the iterative above solution algorithm are currently enabled only for an aerothermal (thermostructure-thermofluid) analysis in which the thermostructure problem is singular. In that case, *A* represents the matrix of the thermostructure subsystem and *R* represents a basis of its null space.

Note 1: An equation solver must be specified under the STATICS command.

The input format of this command can be as follows.

```
QSTATICS
```

```
MECH θ tolqs maxqs delta
HEAT θ β   tolqs maxqs
```

θ Underrelaxation factor $0 < \theta \leq 1$.

β Underrelaxation factor $0 < \beta \leq 1$.

`tolqs` Convergence tolerance.

`maxqs` Maximum number of iterations.

`delta` Time-step equivalent-value of the increment between any two iterations. This parameter is taken into account only in: (a) the presence of a user-defined “control.C” file (see ACTUATORS, see SENSORS, see USDD, and see USDF.) , and/or (b) the time-stamp in the output format for quasi-static analysis. It is the equivalent of a

time-step and is used to convert the current iteration into current time. The default value is 0. When `delta` is specified to a non-zero value, θ is automatically set to $\theta = 1$ and convergence is not checked because in that case, convergence is reached in one iteration and it is understood that the purpose of the quasistatics analysis is to solve a series of consecutive static problems whose right-hand sides are set by a "control.C" file.

Next: READMODE, Previous: QSTATICS

80 RANDOM

Command Statement:	**RANDOM**

The `RANDOM` command statement is used to signal that some materials have properties with random values. All random material properties are assumed to have a Gaussian probability distribution.

Note 1 : Currently, an input file can specify only one random material property per group.

Note 2 : To ensure the positivity of the material properties at every realization, the following non-Gaussian model is used

$$\mathrm{MATPROP} = \mathrm{MEAN} + \frac{\mathrm{STDV}}{\sqrt{2}}(\xi^2 - 1)$$

with the constraint $\frac{\mathrm{STDV}}{\sqrt{2}} < \mathrm{MEAN}$. Here ξ denotes a standard normal random variable.

The input format of this command is given below.

```
RANDOM
```

```
GROUP#          MATPROP          MEAN          STDV
```

`GROUP#`	Id of the element group for which the random properties are to be specified (integer).
`MATPROP#`	Material property for which the random properties are to be specified (string). Currently, this keyword can take any of the following values: `E` (Young's modulus), `A` (cross section of a bar or beam element), and `kx`, `ky`, and `kz` (linear spring coefficients (see MATERIAL).
`MEAN`	Mean value of the non deterministic `MATPROP` (real). This value overwrites the deterministic value of `MATPROP` specified under the `MATERIAL` command for the elements in the corresponding `GROUP#`.
`STDV`	Standard deviation of the non deterministic `MATPROP` (real).

Next: REBUILD, Previous: RANDOM

81 READING EIGEN MODES

Command Statement:	**READMODE**

The READMODE command can be used for inputing a modal basis and modal initial conditions (see IDISPLACEMENTS, IVELOCITIES) for dynamic computations (see DYNAMICS), or performing a "Multi-Ping-Pong" analysis (see AERO).

```
READMODE pathandfilename
```

pathandfilename — Name of the file containing the modes to be read (characters). The format of this file is as follows. The first line contains the number of modes to be read. The second line specifies the number of nodes of the underlying mesh. The modes follow afterwards. Each mode can be specified by a frequency (number of cycles per second) on a separate line and a number of lines equal to the number of nodes, each containing the 3 or 6 values of the mode for the 3 translational or all 6 degrees of freedom of that node. Alternatively, each mode can also be specified by a frequency (number of cycles per second) on a separate line and a number of lines equal to the number of nodes, each containing the node number followed by 3 or 6 values of the mode for the 3 translational or all 6 degrees of freedom of that node.

Next: RENUMBERING, Previous: READMODE

82 REBUILD

Command Statement:	**REBUILD**

The REBUILD command statement is used only in nonlinear analyses to specify when to rebuild the tangent stiffness matrix.

```
REBUILD nsteps
```

nsteps — Specifies the number of Newton iterations after which the tangent stiffness matrix is to be continuously rebuilt/updated (integer). The default is 1.

Next: RESTART, Previous: REBUILD

83 RENUMBERING

Command Statement:	**RENUMBERING**

The RENUMBERING command statement is used to specify the type of node renumbering to be performed on the mesh for either or both of the skyline and sparse solvers. The input format is given below.

```
RENUMBERING
```

```
TYPE
```

TYPE	For FEM, the “rcm” and “sloan” schemes are available for the skyline solvers, and the “esmond” (minimal degree ordering) and “metis” (recursive spectral bisection ordering) are available for the (esmond) sparse solver. When a FETI solver is used, two renumbering schemes can be specified if a skyline as well as a sparse solver are to be used by this FETI solver. In this case, the two renumbering schemes can be specified in any order, each on a separate line (string).

Next: TRBM, Previous: RENUMBERING

84 RESTART

Command Statement:	**RESTART**

The RESTART command is used to request saving computational data in a RESTART file in order to enable later the restart of a dynamics simulation, and/or initializing a dynamics computation using computational results previously saved in a RESTART file (restart data). The processes of creating and reading from a RESTART file are specified on separate lines using different syntaxes.

Note 1: In addition to the time-instances implied by the specified value of the parameter INCREMENT, the restart data is also saved at the of a simulation if **FEM** exits gracefully.

The syntax for invoking this command is given below.

```
PATHANDFILENAME1 INCREMENT
PATHANDFILENAME2 EXTENSION
```

INCREMENT	An integer number that specifies at every how many time-integration steps the “rcfem.restart” file will be updated. For an updating restart, this number specifies at every how many iterations the file “restart.upd” will be updated.
PATHANDFILENAME1 INCREMENT	This command, which can be combined with the following one, instructs **FEM** to save/overwrite the restart data into the specified file PATHANDFILENAME1 (string) every INCREMENT (integer) time-steps. An example using RESTART can be found in FEM.d/fem_examples/Restart.d/
PATHANDFILENAME2 EXTENSION	This command, which can be combined with the previous one, instructs FEM to read the restart data from the specified file PATHANDFILENAME2 (string) and append the extension EXTENSION (string) (e.g. “.2”) to all output filenames specified under the command OUTPUT. An example using RESTART can be found in FEM.d/fem_examples/Restart.d/

Next: RBMFILTER, Previous: RESTART

85 RIGID BODY (AND OTHER ZERO ENERGY) MODES

Command Statement:	**TRBM**

The TRBM command is used to specify a tolerance for monitoring small pivots during the factorization of a matrix. “Small” pivots are deemed to be zero pivots. Therefore, they are associated with rigid body modes (or zero energy modes in general). When this command is specified in the **FEM** input file and a direct solver is specified under the STATICS command, the generalized inverse of the stiffness, conductivity, mass (when an initial acceleration is to be computed to satisfy the governing equation (see DYNAMICS)), or other relevant matrix is computed in factored form by eliminating the equation associated with a deemed zero pivot and setting the corresponding unknown to zero. This command can be used together with the GRBM command.

Note 1: See GRBM for an alternative option for analyzing singular systems.

Note 2: If both the GRBM and TRBM commands are specified in the same **FEM** input file, the one specified last is chosen. However, if that one is TRBM, it does not de-activate the GRBM command but simply overwrites the value of its second tolerance, VALUE_2, by the value of the tolerance specified under TRBM.

TRBM

VALUE

VALUE	Tolerance for monitoring the zero pivots of a matrix during its factorization. The default value is 1.0e-16 (real).

Next: SENSORS, Previous: TRBM

86 RIGID BODY MODES FILTER

Command Statement:	**RBMFILTER**

This command is suitable for structure, aeroelastic, and aerothermoelastic simulations involving an unrestrained (or partially restrained) object when: 1) the effect of the rigid body modes is not sought-after, 2) prescribing Dirichlet (displacement) boundary conditions to remove the rigid body modes is not desirable. When this command is specified in the input file and the structural problem contains rigid body modes, a non-trivial projector is constructed. In a static or quasistatic analysis, the transpose of this projector is applied to the (possibly variable) right hand-side to make sure that the external load is self-equilibrated and therefore admissible. In a dynamic analysis: a) this projector is applied to the initial solution (initial displacement *and* velocity fields), and b) at each time-step, the transpose of this projector is applied to the (time-dependent) right- hand side of the equation being solved. In all cases, the net effect is to work with (and output) a solution where the structural rigid body modes have been filtered out.

The constructed projector can be written as $P = I - R(R^T Q R)^{-1} R^T Q$ where I is the identity matrix, R is the matrix of the rigid body modes of the structure (always computed with the GRBM method),and Q is either the identity matrix for static analysis, or the mass matrix for quasistatic and dynamic analyses. Hence, in a quasistatic analysis, this command requires the density to be present in the material properties specified under the MATERIAL command. An example input file using this command can be found in FEM.d/fem_examples/Rbm_Filter.d.

Note 1: This command works with all solvers except the FETI type solvers. It is implemented for linear static, linear quasistatic, linear dynamic, and nonlinear dynamic analyses. It is not currently implemented for nonlinear static analysis.

Note 2: This command is not active when a modal dynamic analysis is performed (see DYNAMICS). The reason is that one can achieve the same objective simply by not including the rigid body modes in the input for the READMODE command.

The input format of this command is specified below.

```
RBMFILTER
```

level Leaving this blank is equivalent to choosing the value of 1.

1 This is the default value. In this case, the projector is used as described above, in both linear and nonlinear analyses.

2 This value is applicable only for linear structural dynamics problems. In this case, the projector is also applied at each time-step to the outcome of the solution of the system of dynamic equations. As a result, the rigid body modes are also filtered directly from the displacement and velocity fields. This is a safety mesasure in case the previous option does not work properly, for example, because of some ill-conditioning.

Next: SLOSH, Previous: RBMFILTER

87 SENSORS *S*

Command Statement:	**SENSORS**

The SENSORS command statement is used to specify to FEM the degrees of freedom to be observed and whose structural state is to be passed to the user defined control subroutine “control.C” (see ACTUATORS). An example input file using the SENSORS command can be found in FEM.d/fem_examples/Control.d

Note 1: For nonlinear analyses, sensor information on the velocity and acceleration of a nodal degree of freedom is not currently available for use in the “control.C” file. These are currently passed as zero.

```
SENSORS
```

```
NODE#          DOF#
```

NODE# Node number where the sensor is to be placed (integer).

DOF# Degree of freedom local number where the sensor is to be placed (integer).

Next: SZEM, Previous: SENSORS

88 SLOSHING PROBLEMS

Command Statement:	**SLOSH**

The SLOSH command statement is used to specify the computation of the sloshing modes of an incompressible and inviscid fluid whose free surface is orthogonal to a specified gravity field. The computational approach assumes linear kinematics, is based on a fluid displacement potential, and does not involve the fluid material properties. It requires identifying a free surface using free-surface elements (element type 302 in two dimensions and element type 312 in three dimensions, see TOPOLOGY) but does not require any other boundary condition.

Hence, this command should be used in conjunction with the EIGEN and STATIC commands in order to solve the arising eigenvalue problem.

```
SLOSH
```

```
SLOSH       slgrav
```

slgrav	Magnitude (scalar) of the gravitational acceleration whose direction must be orthogonal to the free surface of the problem. FEM uses this specified value to convert the sloshing eigenvalues to sloshing frequencies (number of cycles per second) for output (real).

Next: STATICS, Previous: SLOSH

89 SLOSHING ZERO ENERGY MODE

Command Statement:	**SZEM**

This command is effective only for a sloshing eigen computation involving a zero energy mode. It is used to request the computation of the constant potential mode using a physics-based algorithm (rather than the tolerance-based algorithm associated with the TRBM command). It should be used in conjunction with the SLOSH and related commands.

```
SZEM
```

Next: SURFACETOPO, Previous: SZEM

90 STATICS

Command Statement:	**STATICS**

The STATICS command statement is used to signal that the following data lines correspond to the selected equation solver to be used within the chosen analysis. The input format is given below.

Note 1: Except when otherwise specified, the solvers proposed below are available only for symmetrical systems of equations.

Note 2: Among all solvers proposed below, only the spooles and mumps sparse direct solvers (with pivoting turned on) and the gmres

and FETI DP iterative solvers are suitable for the solution of indefinite systems. Such systems arise: (1) in conjunction with the HELM, IMPEDANCE, LMPC or TIEDSURFACES command, or the usage of a positive shift in the EIGEN command (see SHIFT in that command), and (2) whenever a structural model includes rigid and/or joint elements (see TOPOLOGY).

Note 3: The Lagrange multiplier method for enforcing constraints leads to an indefinite system of equations. Currently, only the spooles and mumps direct solvers with pivoting turned on and the gmres and FETI DP iterative solvers support this approach for solving constrained systems (see CONSTRAINTS).

Note 4: Among all FETI methods, only FETI DP is maintained for nonlinear structural static and dynamic analyses.

Note 5: When a non deterministic analysis is performed using the intrusive version of the Polynomial Chaos method, only the PCG, BCG, and CR solvers can be selected for solving the resulting large system of equations.

```
STATICS
```

```
METHOD (keyword)
PARAMETERS (keyword and/or value)
```

METHOD

direct	For a skyline direct solver (characters). This is also the default solver (except when the INPC command is used) and is available for frequency-domain acoustic (Helmholtz) problems.
skyline	For a skyline direct solver (characters). This is also the default solver and is available for frequency-domain acoustic (Helmholtz) problems.
sparse	For Esmond's sparse direct solver (characters). This solver is also available for frequency-domain acoustic (Helmholtz) problems.
sgisparse (completely spelled out)	For SGI's sparse direct solver (characters). Runs only on SGI machines.
spooles (completely spelled out)	For the SPOOLES sparse direct solver (characters). Cannot be used for any system of equations with less than 24 unknowns.
spooles pivot (completely spelled out)	For the SPOOLES sparse direct solver with pivoting option turned on (characters).
spooles unsymmetric (completely spelled out)	For the SPOOLES unsymmetric sparse direct solver (characters).
mumps (completely spelled out)	For the MUMPS sparse symmetric positive definite direct solver (characters).
mumps pivot (completely spelled out)	For the MUMPS sparse general symmetric direct solver with pivoting option turned on (characters).
mumps unsymmetric (completely spelled out)	For the MUMPS unsymmetric sparse direct solver (characters).
sgisky (completely spelled out)	For SGI's skyline direct solver (characters). Runs only on SGI machines.
frontal (completely spelled out)	For a frontal direct solver (characters).
pcg	For a preconditioned conjugate gradient solver (characters).
bcg	For a bi-conjugate gradient solver (characters).
cr	For a conjugate residual solver (characters).

gmres	For a generalized minimum residual solver (characters). Applicable to symmetric and unsymmetric systems.
FETI	For a one-level FETI solver (default) (characters).
FETI 1	For a FETI-1 (one-level FETI method) solver (characters).
FETI 2 OLD	For a FETI-2 solver using a “full” coarse problem implementation (characters).
FETI 2 NEW	For a FETI-2 solver using a “sparse” coarse problem implementation (characters).
FETI DP	For a FETI-DP (FETI dual-primal method) solver (characters).
FETI DPH	For a FETI-DPH (FETI dual-primal method with augmentation basis) solver (characters). This solver is to be used only with the IMPEDANCE command or the HELM command. Depending on whether the system to be solved is damped or not, or contains a complex boundary condition or multipoint constraint or not, this solver performs in the real or complex domain.

PARAMETERS (pcg/bcg/cr/gmres solvers, in any order)

precno	Specifies the preconditioner (characters).
0	Unpreconditioned (integer).
1	Diagonal scaling (integer). This option should not be used when solving an indefinite system arising in conjunction with the usage of the LMPC or TIEDSURFACES command, or whenever a structural model includes rigid and/or joint elements (see TOPOLOGY), because in all these cases, the system matrix will contain zero diagonal entries.
2	Incomplete block diagonal scaling (integer). This option is available only for a system of equations resulting from the intrusive version of the Polynomial Chaos method. In this case, the subsystems of equations associated with the diagonal blocks can be solved either using the sparse solver (default), or the FETI DP method with its acceleration for multiple right sides. If the FETI DP method is chosen as a block diagonal solver, then its keyword and the keywords associated with its options should be specified after all other parameters of the main (outer) solver have been specified.
tolpcg	Error tolerance for the convergence of the pcg solver (real).
tolbcg	Error tolerance for the convergence of the bcg solver (real).
tolcr	Error tolerance for the convergence of the cr solver (real).
tolgmres	Error tolerance for the convergence of the gmres solver (real).
maxitr	Maximum number of iterations to be performed (integer).

PARAMETERS (spooles solver, in any order)

spooles_renum	Specifies the renumbering (integer, default value is 0).
0	Best of nested dissection and multisection.
1	Multiple minimum degree.
2	Multisection.
3	Nested dissection.

spooles_scale	Specifies the scaling (integer, default value is 0).
0	No scaling.
1	Symmetric scaling.
spooles_tau	Upper bound on the magnitude of the largest element in L or U when pivoting enabled (real 1.0). If this number is too small the results will be wrong. The default value of 100 is generally safe if scaling is used (see spooles_scale).
spooles_msglvl	Message output level (integer $\geq$ 0, default value is 0).
spooles_maxdomainsize	n /spooles_maxdomainsize is the maximum subgraph size used by SPOOLES orderings, where n is the number of equations in the system to be solved. This parameter is used to control the incomplete nested dissection process. Any subgraph whose weight is less than maxdomainsize is not split further (integer $\geq$ 0, default value is 24).
spooles_maxzeros	n *spooles_maxzeros is the maximum number of zeros allowed in a supernode/front (real > 0 and $\leq$ 1.0, default value is 0.04).
spooles_maxsize	Maximum number of internal columns in supernode/front (integer > 0, default value is 64).

PARAMETERS (mumps solver, in any order)

mumps_icntl #	Value for ICNTL(#), see mumps documentation for specific details (integer).
mumps_cntl #	Value for CNTL(#), see mumps documentation for specific details (real).

PARAMETERS (FETI, FETI-DP, FETI-DPH solvers, in any order)

sparse	Specifies Esmond's sparse direct method as the local (subdomain and Dirichlet preconditioner) solver (characters). The default is a skyline solver which can also be invoked by specifying skyline instead of sparse on this line. Note also that there is another mechanism for specifying this option (see below).
local_solver	
skyline	Selects the skyline direct method as the local solver (subdomain and Dirichlet preconditioner problems) (characters). This is also the default choice.
sparse	Selects Esmond's sparse direct method as the local solver (characters).
spooles	Selects the SPOOLES sparse direct method as the local solver (characters).
spooles pivot	Selects the SPOOLES sparse direct method as the local solver with pivoting option turned on (characters).
mumps	Selects the MUMPS sparse direct method as the local solver (characters).
mumps pivot	Selects the MUMPS sparse direct method as the local solver with pivoting option turned on (characters).
coarse_solver	
blockskyline	Selects the block-skyline direct method as the coarse solver. This is also the default choice (characters).
skyline	Selects the skyline direct method as the coarse solver (characters).

sparse	Selects Esmond's (sequential) sparse direct method as the coarse solver (characters).
psparse	Selects the parallel sparse direct method as the coarse solver (characters).
spooles	Selects the SPOOLES sparse direct method as the coarse solver (characters).
spooles pivot	Selects the SPOOLES sparse direct method as the local solver with pivoting option turned on (characters).
mumps	Selects the MUMPS sparse direct method as the coarse solver (characters).
mumps pivot	Selects the MUMPS sparse direct method as the local solver with pivoting option turned on (characters).
precno	Specifies the local preconditioner (integer or characters).
0 or noprec	Unpreconditioned (integer).
1 or lumped	Lumped preconditioner (integer). This is also the default preconditioner.
2 or dirichlet	Dirichlet preconditioner (integer).
projector	
1	Identity based projector (integer). This is also the default projector.
2	Preconditioner based projector (integer). Uses the selected preconditioner for building the so-called Q matrix.
4	Superlumped projector (integer).
scaling	
1 or stiffness	Stiffness based scaling (integer). This is also the default scaling procedure.
2 or topology	Topology (subdomain connectivity) based scaling (integer).
version	
1	The FETI (also known as FETI-1) method (integer).
2	The two-level FETI (also known as FETI-2) method (integer).
nocoarse	This option is only for dynamics. When specified, the FETI algorithm is executed without any “coarse grid”.
corners	This keyword is useful only for the FETI-2, FETI-DP, and FETI-DPH methods. It specifies the treatments of the corners and corner dofs for the contruction by these iterative methods of their respective coarse problems. For the FETI-2 method, the user can define both the corner and corner dof selection algorithms. For the FETI-DP and FETI-DPH methods, the corners are automatically chosen by **FEM** but the user can still control the number of dofs at these corners.
cp3	In that case, a corner is defined by FETI-2 as a crosspoint, and only the three active translational dof attached at each corner node are included in the construction of the FETI-2 or FETI-DP and FETI-DPH coarse problems. This is the default value for nodes with 3 dofs. Here, a crosspoint is defined as a point that belongs to more than four subdomains.

`cp6`	In that case, a corner is defined by FETI-2 as a crosspoint, and all six active dof attached at each corner node are included in the construction of the FETI-2 or FETI-DP and FETI-DPH coarse problems. This is the default value for nodes with 6 dofs. Here, a crosspoint is defined as a point that belongs to more than two subdomains.
`be3`	In that case, a corner is defined by FETI-2 as the beginning or end of and edge, and only the three active translational dof attached at each corner node are included in the construction of the FETI-2 or FETI-DP and FETI-DPH coarse problems. Note that a crosspoint is also the beginning or end of an edge.
`be6`	In that case, a corner is defined by FETI-2 either as the beginning or end of and edge, and all six active dof attached at each corner node are included in the construction of the FETI-2 or FETI-DP and FETI-DPH coarse problems. Note that a crosspoint is also the beginning or end of an edge.
`augment`	This keyword is useful only for the FETI-DP and FETI-DPH methods. It specifies the augmentation of the “coarse grid” by various methods defined by the following commands.
`EdgeGs trans/all`	In that case, the “coarse grid” is augmented using extra equations generated by the rigid body modes (rbms) of the subdomain interfaces, and organized edge-by-edge. Here an edge refers to an interface between two subdomains and not to the usual geometric edge. The “trans/all” option denotes the equation type where “trans” refers to the translational rbms, and “all” refers to both the translational and roational rbms applied per edge.
`Gs trans/all`	In that case, the “coarse grid” is augmented using extra equations generated by the traces of the subdomain rigid body modes (rbms) on the subdomain interface boundaries, and organized subdomain-by-subdomain. The “trans/all” option denotes the equation type where “trans” refers to the translational rbms, and “all” refers to both the translational and roational rbms.
`WeightedEdgeGs trans/all`	In that case, the “coarse grid” is augmented using extra equations generated by the weighted rigid body modes (rbms) of the subdomain interfaces, and organized edge-by-edge. Here an edge refers to an interface between two subdomains and not to the usual geometric edge. The “trans/all” option denotes the equation type where “trans” refers to the translational rbms, and “all” refers to both the translational and roational rbms applied per edge. The weights are similar to those used for scaling the residuals. Hence, they are based on stiffness considerations if the `scaling` option is set to `stiffness`, or on topological considerations if the `scaling` option is set to `topology`.
`EdgeWs numdir`	This option is exclusive to the FETI-DPH solver and can be combined with the option `EdgeGs`. It augments the corner-based coarse problem of the FETI-DPH algorithm with extra equations generated by the free-space solutions of the frequency-

domain acoustic or elastodynamic (or modelled dynamic shell) equation — these are real cosine and sine waves of arbitrary directions — and organized edge-by-edge. Here an edge refers to an interface between two subdomains and not to the usual geometric edge. There are as many free-space solutions to be considered as there are wave directions to be considered. The parameter `numdir` specifies the number of desired directions and therefore controls the total number of such wavy augmentation modes (integer). Its default value is 0. The exact number of augmentation modes per interface edge is equal to $2\times$ `numdir` for two- and three-dimensional frequency-domain acoustic problems, $4\times$ `numdir` for two-dimensional elastodynamic (or modelled dynamic shell) problems, and $6\times$ `numdir` for three-dimensional elastodynamic (or modelled dynamic shell) problems. In the latter case, the factor 6 comes from the fact that for each direction, there are 2 shear waves and 1 pressure wave, and each of these three waves has a cosine mode as well as a sine mode. The implemented directions are chosen according to the following scheme. In two dimensions, a sector is discretized into *n* sectors, with *n* an even integer. A direction is defined by connecting the center of the circle to a point on the circle delimiting a sector. Since both the cosine and sine modes are included, only one direction for each pair of opposite directions needs be retained, which results in a total of *n*/2 directions. Hence, *n* is chosen to be 2*`numdir`. In three dimensions, a cube is discretized into $n \times n \times n$ points. A direction is defined by connecting the center of the cube to a point lying on a face of the cube. Since both the cosine and sine modes are included, only one direction for each pair of opposite directions needs be retained, which results in a total of $n^2 + 4(n-1)(n/2-1) + 2(n-1)mod(n,2)$ directions. Hence, *n* is chosen so that $n^2 + 4(n-1)(n/2-1) + 2(n-1)mod(n,2)$ is as close as possible to `numdir`, with $n^2 + 4(n-1)(n/2-1) + 2(n-1)mod(n,2) \geq$ `numdir`.

`interf_solver` — This option is currently available only for the FETI-DP and FETI-DPH solvers. It speficies the Krylov method to be used with these algorithms for solving the interface problem (characters).

- `CG` — Turns on the CG algorithm as an interface problem solver. This is the default value of `interf_solver`.
- `CGAL` — Turns on Dostal's Augmented Lagrangian CG algorithm with adaptive precision control as an interface problem solver.
- `GMRES` — Turns on the GMRES algorithm as an interface problem solver.
- `GCR` — Turns on the GCR algorithm as an interface problem solver. This option is more computationally efficient when performing frequency or wavenumber sweeps (see IMPEDANCE, and/or see HELMHOLTZ) as the GCR algorithm is optimized for the systems with multiple right sides that arise during the reconstruction of solutions in frequency/ wavenumber bands.

`orthotol` This option is currently available only for the FETI-DPH solver. It specifies the tolerance to be used for filtering out “small” vectors during the local Gram-Schmidt-like orthogonalization of the augmentation vectors. The default value is 1.0e-02.

`mpc_type` This keyword is useful only for the FETI-DP and FETI-DPH methods. It specifies the algorithm to be used for handling potential multipoint constraints (MPCs) (characters). The default is `primal` when the number of MPCs is less or equal to 1000, and `dual` otherwise.

- `dual` In this case, the MPCs are enforced by Lagrange multipliers and are satisfied only at convergence (characters).
- `primal` In this case, the MPCs are put in the coarse problem and are satisfied at every iteration (characters).

`mpc_precn` This keyword is useful only for the FETI-DP and FETI-DPH methods, in the presence of MPCs, and when `mpc\_type` is set to `dual` (characters). The default is `tblock` when running a single thread process, and `full` otherwise.

- `full` In this case, the CC^T matrix, where *C* denotes the constraint matrix, is treated as a single block matrix (characters).
- `tblock` In this case, the potential algebraic block-structure of the CC^T matrix, is exploited (characters).
- `sblock` In this case, the subdomain-structure of the CC^T matrix, is exploited (characters).
- `mblock` In this case, the mortar-interface-structure of the CC^T matrix, is exploited (characters).
- `diag` In this case, the CC^T matrix is approximated by its diagonal (characters).

`mpc_scaling`

- `1 or stiffness` Stiffness based scaling (integer).
- `2 or topology` Topology (subdomain connectivity) based scaling (integer). This is also the default scaling procedure.

`cct_solver`

- `skyline` Selects the skyline direct method as the coarse solver. This is also the default choice (characters).
- `sparse` Selects Esmond's (sequential) sparse direct method as the coarse solver (characters).

`cct_tol` Specifies the tolerance to be used for detecting singularities in the solution of the CC^T problems of the FETI-DP and FETI-DPH methods in the presence of MPCs (real). The default is 1.0E-12.

`kryprec` This option is only for nonlinear problems. It turns on the Krylov preconditioner enrichment for the solution of nearby problems in a Newton method (characters).

1	Turns on the Krylov preconditioner and gives to the maxorth parameter (see below) the scope of the entire nonlinear analysis.
2	Turns on the Krylov preconditioner and gives to the maxorth parameter (see below) the scope of a load-step in a nonlinear analysis.
global_cor_rbm_tol	Specifies the tolerance to be used for detecting singularities in the solution of the corner-based coarse problems of the FETI-DP and FETI-DPH methods (real). The default is 1.0E-6.
global_rbm_tol	Specifies the tolerance to be used for detecting the global rigid body modes when using a FETI method (real). The default is 1.0E-6.
maxorth	Specifies the maximum total number of reorthogonalization vectors used for accelerating FETI and/or enriching its chosen preconditioner during a simulation (integer). For nonlinear problems, if the kryprec option is not used, this total is to be understood per Newton iteration, as the orthogonalization vectors are flushed when the tangent operator is rebuilt. Also for nonlinear problems, if the kryprec option is used, priority in the accumulation of vectors is given to enriching the preconditioner. The default is maxitr (see below). To turn-off this reorthogonalization option, set maxorth to 1.
tolfeti	Error tolerance for the convergence of the feti solver (real). The default value is 1.0e-06.
maxitr	Maximum number of iterations to be performed (integer). The default value is 500.

Next: TETT, Previous: STATICS

91 SURFACE TOPOLOGY

Command Statement:	**SURFACETOPO**

The SURFACETOPO command statement can be used to define a surface on a body by describing its faceted connectivity. Such a surface can be paired with another similar surface using another command statement in order to define a surface-to-surface interaction (see TIEDSURFACES, CONTACTSURFACES, FSINTERFACE, AERO), or can be referred to by another command statement to define a surface boundary condition (see PRESSURE, FORCES and DISPLACEMENTS).

This command can be repeated as many times as there are surfaces. Its format is as follows.

```
SURFACETOPO        ID_NUMBER        SHELL_THICKNESS        t
```

ID_NUMBER	Surface id number (integer).
SHELL_THICKNESS	Optional keyword indicating that the surface identified by ID_NUMBER defines the midplane of a shell (string).
t	Thickness of shell surface (real).

```
FACE#        FACETYPE        CONNECTIVITY_NODES
```

FACE#	Face id number whose type and connectivity are to be specified (integer).
FACETYPE	
1	4-node quadrilateral.
2	8-node quadrilateral.
3	3-node triangle.
4	6-node triangle.
5	9-node quadrilateral.
6	12-node quadrilateral.
7	10-node triangle.

Next: THERMOE, Previous: SURFACETOPO

92 THERMAL EXPANSION TEMPERATURE TABLE

Command Statement:	**TETT**

The TETT command statement can be used to describe the evolution of the coefficient of thermal expansion with temperature, for a given material. This evolution can be specified here in a curve (or table) defined by pairs of temperature and coefficient of thermal expansion values. Linear interpolation is used for “in between” points, and the extrema values are adopted for “outside” points. Several curves can be specified, one after the other. Each curve is identified by an “id number” as described below.

```
TETT
```

```
CURVE curve_id
T_1   TE_1
.     .
.     .
.     .
T_n   TE_n
```

CURVE	
curve_id	“Id number” for the following curve (or table) (integer).
T_1	A specified temperature value (float).
TE_1	A specified coefficient of thermal expansion value at temperature T_1 (float).

Next: THERMOH, Previous: TETT

93 THERMOELASTICITY

Command Statement:	**THERMOE**

The THERMOE command statement is used to indicate that FEM is to interact with FEM-HEAT to perform a *transient* thermoelastic (thermostructure-structure vibration) *one-way* coupled simulation.

The syntax for invoking this option is given below.

```
THERMOE
```

Next: TIEDSURFACES, Previous: THERMOE

94 THERMOELASTICITY: FEM-HEAT (FEM-HEAT Only)

Command Statement:	**THERMOH**

The THERMOH command statement is used to indicate that FEM-HEAT is to interact with FEM to perform a *transient* thermoelastic (thermostructure-structure vibration) *one-way* coupled simulation.

The syntax for invoking this option is given below.

```
THERMOH
```

Next: TOPOLOGY, Previous: THERMOH

95 TIED SURFACES

Command Statement:	**TIEDSURFACES**

The TIEDSURFACES command statement can be used to tie — that is, enforce perfect contact between — pairs of surfaces defined using the SURFACETOPO command. Surface interactions are detected using ACME's search module.

Note 1: For the enforcement of tied surface constraints by the Lagrange multiplier method in explicit dynamic analysis, the discrete kinematic constraint equations are defined using a node-to-segment approach and the constraint forces are computed using ACME's enforcement module.

Note 2: For the enforcement of tied surface constraints by either the Lagrange multiplier method in all but explicit dynamic analyses, or the penalty or elimination method in all analyses, the discrete kinematic constraint equations are defined using a segment-to-segment (mortar) approach.

Note 3: The enforcement of tied surface constraints by the Lagrange multiplier method in all but explicit dynamic analyses is supported only by the FETI-DP family of solvers, the GMRES solver, and the SPOOLES and MUMPS direct sparse solvers with pivoting enabled (see STATICS).

```
TIEDSURFACES
```

```
SURF_PAIR_ID#     MASTER     SLAVE
```

or, for statics, frequency response, eigenvalue problems and dynamics with an implicit scheme

```
SURF_PAIR_ID#     MASTER     SLAVE     CONSTRAINT_METHOD     MORTAR_TYPE     NORMAL_TOL     TANGENTIAL_TOL
```

or, for dynamics with an explicit scheme

```
SURF_PAIR_ID#     MASTER     SLAVE     CONSTRAINT_METHOD     KPART_TYPE     NORMAL_TOL     TANGENTIAL_TOL     NUM_ITER
CONVERG_TOL
```

	SURF_PAIR_ID#	Id number of the surface pair to be described (integer).
	MASTER	Identification of the master (mortar method) surface (see SURFACETOPO) (integer).
	SLAVE	Identification of the slave (mortar method) surface (see SURFACETOPO) (integer).
	CONSTRAINT_METHOD	Method for enforcing the associated constraints (characters). The default method is set in CONSTRAINTS and used whenever this entry is omitted.
	multipliers	The Lagrange multiplier method.
	elimination	The elimination method.
	penalty	The penalty method. The parameter beta should be a large positive number, typically of the order
[beta]		of 10^8 (no default value is provided).
	MORTAR_TYPE	Mortar type: 0 = standard, 1 = dual, default value is 0 (integer).
	NORMAL_TOL	Normal search tolerance used by ACME to identify interactions, default value is 0.1 (float) (see Figs. 1.2 and 1.3 in Section 1.3 of ACME's User Reference Manual).
	TANGENTIAL_TOL	Tangential search tolerance used by ACME to identify interactions, default value is 0.001 (float) (see Figs. 1.2 and 1.3 in Section 1.3 of ACME's User Reference Manual).
	KPART_TYPE	Kinematic partitioning type for explicit dynamics: 0 = fixed, 1 = automatic, default value is 0 (integer).
	NUM_ITER	Maximum number of predictor-corrector iterations to be performed at each time step for explicit dynamics, default value is 5 (integer).
	CONVERG_TOL	Convergence tolerance of the predictor-correctior iteration loop for explicit dynamics, default value is 1.0e-10 (float).

Next: USDF, Previous: TIEDSURFACES

96 TOPOLOGY *S*

```
Command Statement:          TOPOLOGY
```

The TOPOLOGY command statement is used to signal that the following data lines correspond to the connectivity and type of each

element. The input format is given below. There should be as many lines as there are elements in the system.

Note 1: In the absence of the NONLINEAR command in the input file, all element types imply linear kinematics and a linear elastic material even when otherwise specified.

Note 2: In the absence of the NONLINEAR command in the input file or the absence of the keyword nlmat under that command, all elements imply a linearly elastic material even when otherwise specified. The properties of linearly elastic materials are specified under the MATERIAL command.

Note 3: Materially and geometrically + materially nonlinear elements are supported only by **FEM**.

Note 4: The complete description of elements with a nonlinear material can be performed using both the MATERIAL and MATLAW commands. The usage of such elements requires the specification in the input file of the NONLINEAR command, and the presence of the nlmat keyword under that command.

Note 5: FEM supports a run-time generation of beam frames (see EFRAMES) that is activated when a third node in the local *x*-*z* plane is found in the definition of a beam element within this TOPOLOGY command. The only requirement for the third node is that it does not be colinear with the other two beam nodes. Using this third node option releaves the user from specifying the EFRAMES command. An example illustrating the third node option can be found in FEM.d/fem_examples/Third_Node.d.

Note 6: Geometric stiffnesses are currently implemented only for the truss, beam, shell, 4-noded tetrahedron, 8-noded brick, and spring elements. Some of these are based on the corotational method, some on the Lagrangian method. Mixing the corresponding geometric stiffness matrices is not a problem.

Note 7: Thermal loads (see TETT) are currently implemented only for the Euler beam, 4-noded plane stress/strain, 4-noded as well as 10-noded tetrahedron, 8-noded brick, and 6-noded penta (prism with triangular cross section) elements.

Note 8: For thermal analysis, **FEM** constructs the contribution of convection effects to the right-hand side vector of the resulting system of equations using the CONVECTION command. However, it constructs the contribution of convection effects to the stiffness matrix using the “boundary convection” elements (type 47, 48 and 49) which must be superposed to the edges or faces of existing thermal elements. The data for these boundary convection elements (depth, area, convection coefficients and ambient temperature) must be specified in the MATERIAL command.

Note 9: Currently, only the SPOOLES and MUMPS sparse direct solvers and the FETI-DP(H) solvers can handle rigid elements.

Note 10: Rigid elements are internally converted by **FEM** to linear multipoint constraints (see LMPC) in the following cases: (a)during an eigen analysis if the GRBM (see GRBM) is present in the input file, except for element types 66, 73 and 74, (b) during a nonlinear analysis, except for element type 65, and (c) when the FETI-DP equation solver is specified.^M

Note 11: For DEM (Discontinuous Enrichment Method) and DGM (Discontinuous Galerkin Method) elements, Q, T, and H designate a quadrialteral, a triangle in two dimensions and a tetrahedron in three dimensions, and an hexahedron, respectively. In the notation Q-X-Y, T-X-Y, and H-X-Y, X denotes the number of enrichment functions in the element and Y denotes the number of Lagrange multiplier degrees of freedom per edge or face of the element. In the notation X1xX2-Y for elastodynamics, X1 denotes the number of directions of the plane wave enrichment functions, X2 denotes the number of enrichment functions per wave direction (X2 = 2 (one pressure and one shear wave) in two dimensions and X2 = 3 (one pressure and two shear waves) in three dimensions), and Y denotes the number of Lagrange multiplier degrees of freedom per edge of face of the element. The connectivity (geometry and local node numbering) of all Q, T, and H DEM and DGM elements is that of Q, T, and H higher-order isoparametric elements, respectively. Hence, for each DEM element, the degree of the polynomial field can be deduced from the number of nodes of that element.

Note 12: The fluid elements (type 301, 302, 311, 312, 321, and 331) are active only in the following cases: (1) a mass computation using the MASS command, (2) a sloshing eigen computation using the SLOSH and related commands, (3) a hydroelastic eigenvalue computation using the EIGEN, HEFRS, HEFSB and related commands.

ELEMENT#	ELEMENT_TYPE	CONNECTIVITY_NODES

ELEMENT#	Element identification number whose type and connectivity are to be specified (integer). A (*) indicates the element is not yet fully implemented and/or validated.
ELEMENT_TYPE:	M=Mechanics - A=Acoustic (Helmholtz or Time-Domain) - H=Thermal - C=Coupled Thermoelastic - F=Fluid
1	M: 3d truss (bar) element with 3 dof/node. **Note that only a lumped mass matrix is available for this element**.
2	M: 2d 4-node quad element with 2 dof/node. **Note that only a lumped mass matrix is available for this element**.
3	H: 3d 4-node quad element with 1 dof/node.
4	M: 2d 3-node triangular element with 2 dof/node. **Note that only a lumped mass matrix is available for this element**.
6	M: 3d Bernoulli beam element with 6 dof/node.
7	M: 3d Timoshenko beam element with 6 dof/node. **Note that only a lumped mass matrix is available for this element**.
8	M: 3d 3-node triangular AQR shell element with 6 dof/node. **Note that only a lumped mass matrix is available for this element**.
88	M: 3d 4-node shell element element with 6 dof/node that splits into 2 triangular shell elements of type 8. **Note that only a lumped mass matrix is available for this element**.
9	H: 3d 2-node lineal element with 1 dof/node.
10	H: 2d 4-node quad element with 1 dof/node.
11	M: 3d torsion spring element with 3 dof/node.
12	C: 3d thermomelastic truss element with 4 dof/node.

13	C: 2d thermoelastic quad element with 3 dof/node.
14	C: 3d thermoelastic Bernoulli beam element with 7 dof/node.
16	M: 3d Belytschko-Tsay quadrilateral shell element with 6 dof/node and 1-point quadrature rule. **Currently, this element is available only for explicit dynamic analyses**.
17	M: 3d 8-node brick element with 3 dof/node.
18	M: 3d 4-node tetrahedra element with 6 dof/node. **Note that only a lumped mass matrix is available for this element**.
19	M: 3d 3-node triangular membrane element with 6 dof/node. (Only in-plane and drilling stiffnesses). **Note that only a lumped mass matrix is available for this element**.
20	M: 3d 3-node triangular composite or orthotropic shell element with 6 dof/node. **Note that only a lumped mass matrix is available for this element**.
2020	M: 3d 4-node composite or orthotropic shell element with 6 dof/node. **Note that only a lumped mass matrix is available for this element**.
21	M: 3d translational spring-link element with 3 dof/node.
22	M: 3d torsion spring-link element with 3 dof/node.
23	M: 3d 4-node tetrahedral element with 3 dof/node.
24	M: 3d 6-node pentahedral element (prism with triangular cross section) with 3 dof/node.
25	M: 3d 10-node tetrahedral element with 3 dof/node.
30	A: 2d 4-node quad element with 1 dof/node.
31	A: 2d 4-node quad GLS element with 1 dof/node.
32*	A: 2d 8-node quad element with 1 dof/node.
33*	A: 2d 4-node quad bubble element with 1 dof/node.
34*	A: 2d 4-node quad two-level bubble element with 1 dof/node.
35	A: 2d 3-node triangular element with 1 dof/node.
36	A: 2d 3-node triangular GLS element with 1 dof/node.
38*	A: 2d 6-node triangular element with 1 dof/node.
40	A: 3d 4-node tetrahedral element with 1 dof/node.
41	A: 3d 4-node tetrahedral GLS element with 1 dof/node.
42	A: 3d 10-node tetrahedral element with 1 dof/node.
44	A: 3d 8-node brick GLS element with 1 dof/node.
45*	A: 3d 8-node brick element with 1 dof/node.
46	H: 3d 3-node triangular heat conduction element.
4646	H: 3d 4-node quadrilateral heat conduction element that splits into 2 triangular elements of type 46.
47	H: 3d 2-node lineal element with 1 dof/node for boundary convection (see Note 8).
48	H: 3d 4-node quad element with 1 dof/node for boundary convection (see Note 8).
49	H: 3d 3-node triangular element with 1 dof/node for boundary convection (see Note 8).
50	H: 3d 4-node, or 10-node, or 20-node, or 35-node tetrahedral element with 1 dof/node. **These elements use a special local node-numbering for connectivity which goes line-by-line**.
51	H: 3d 8-node brick element with 1 dof/node.
52*	M: 3d 6-node triangular shell element.

53	H: 2d 3-node triangular heat conduction element.
56	H: 3d 2-node heat radiation element.
57	H: 3d 3-node triangular heat radiation element.
58	H: 3d 4-node quadrilateral heat radiation element.
65	M: 3d 2-node rigid truss (bar) element; enforces constant length of the element.
66	M: 3d 2-node rigid beam element; enforces constant length of the element, equal rotations of the cross sections at its two nodes, and other constraints between its rotational and translational dofs.
67	M: 3d 2-node rigid link (translational and rotational); for each dof at one node, enforces equality to the corresponding dof at the other node.
68	M: 3d 2-node rigid translational link; for each translational dof at one node, enforces equality to the corresponding dof at the other node.
69	M: 3d 2-node rigid rotational link; for each rotational dof at one node, enforces equality to the corresponding dof at the other node.
70	M: 3d 8-node rigid brick element with 3 dof/node; enforces constant distance between each pair of its nodes (special case of element type 71).
71	M: 3d rigid element (lineal, plane, or solid) with 3 dof/node and anywhere from 3 to 32 nodes per element; enforces constant distance between each pair of its nodes.
72	M: 3d 20-node brick element.
73	M: 3d 3-node rigid shell element; equivalent to three rigid beam elements, each defined by an edge of the element.
74	M: 3d rigid element (lineal, plane or solid) with 6 dof/node and anywhere from 3 to 32 nodes per element; enforces constant distance between each pair of its nodes, equal values of the rotational dofs at all nodes, and other constraints to enforce overall rigidity of the element.
81	H: 2d 4-node contact resistance thermal element with 1 dof/node that can be inserted between two thermal elements of type = 10.
82	H: 3d 8-node contact resistance thermal element with 1 dof/node that can be inserted between two thermal elements of type = 51.
83	H: 3d 6-node contact resistance thermal element with 1 dof/node that can be inserted between two thermal elements of type = 50 with 4 nodes each.
84	H: 2d 3-node triangular bulk fluid (thermal) element with 1 dof/node. The first node appearing in the connectivity list of this element must be inside the bulk fluid and the other two nodes must be located at the interface boundary between the bulk fluid and the surrounding material. The temperature at the first node is by definition the average temperature of the bulk fluid.
85	H: 3d 4-node tetrahedral bulk fluid (thermal) element with 1 dof/node. The first node appearing in the connectivity list of this element must be inside the bulk fluid and the other three nodes must be located at the interface boundary between the bulk fluid and the surrounding material. The temperature at the first node is by definition the average temperature of the bulk fluid.
86	H: 3d 5-node pyramidal bulk fluid (thermal) element with 1 dof/node. The first node appearing in the connectivity list of this element must be inside the bulk fluid and the other four nodes must be located at the interface boundary between the bulk fluid and the surrounding material. The temperature at the first node is by definition the average temperature of the bulk fluid.
90	A: 3d 6-node wedge element with 1 dof/node.
91	M: 3d 32-node serendipity brick element with 3 dof/node.

92	M: 3d 26-node serendipity wedge element with 3 dof/node.
93	A: 3d 32-node serendipity brick element with 1 dof/node.
94	A: 3d 26-node serendipity wedge element with 1 dof/node.
95	A: 3d 8-node, or 27-node, or 64-node, or 125-node hexahedral element with 1 dof/node. **These elements use a special local node-numbering for connectivity which goes line-by-line**. They also support the PML (Perfectly Matching Layer) computational technology.
96	A: 3d 4-node, or 10-node, or 20-node, or 35-node tetrahedral element with 1 dof/node. **These elements use a special local node-numbering for connectivity which goes line-by-line**. They also support the PML (Perfectly Matching Layer) computational technology.
97	M: 3d 15-node wedge element with 3 dof/node.
98	A: 2d 4-node, or 9-node, or 16-node, or 25-node quadrilateral element with 1 dof/node. **These elements use a special local node-numbering for connectivity which goes line-by-line**. They also support the PML (Perfectly Matching Layer) computational technology.
99	A: 2d 3-node, or 6-node, or 10-node triangular element with 1 dof/node. **These elements use a special local node-numbering for connectivity which goes line-by-line**. They also support the PML (Perfectly Matching Layer) computational technology.
100	M: 2d 4-node, or 9-node, or 16-node, or 25-node quadrilateral element with 3 dof/node. **These elements use a special local node-numbering for connectivity which goes line-by-line**.
101	M: 2d 3-node, or 6-node, or 10-node triangular element with 3 dof/node. **These elements use a special local node-numbering for connectivity which goes line-by-line**.
102	M: 3d 8-node, or 27-node, or 64-node, or 125-node hexahedral element with 3 dof/node. **These elements use a special local node-numbering for connectivity which goes line-by-line**.
103	M: 3d 4-node, or 10-node, or 20-node, or 35-node tetrahedral element with 3 dof/node. **These elements use a special local node-numbering for connectivity which goes line-by-line**.
105	A: 3d 8-node, or 27-node, or 64-node, or 125-node spectral hexahedral element with 1 dof/node. **These spectral elements use a special local node-numbering for connectivity which goes line-by-line**.
108	A: 2d 4-node, or 9-node, or 16-node, or 25-node spectral quadrilateral element with 1 dof/node. **These spectral elements use a special local node-numbering for connectivity which goes line-by-line**.
111	M: 3d fabric truss element with 3 dof/node. **Note that only a lumped mass matrix is available for this element**.
120	M: 3d 2-node spherical joint element with 3 dof/node.
122	M: 3d 2-node universal joint element with 6 dof/node. This element has a co-rotating reference frame attached to each of its two nodes. Both frames have the same orientation in the underformed configuration and therefore can be viewed as far as input is concerned as the same frame. The orientation of this frame in the undeformed configuration must be specified under EFRAMES, using the same format as for a beam element. Its y axis defines the orientation in the undeformed configuration of the y axis of the co-rotating reference frame at node 2. Its z axis defines the orientation in the undeformed configuration of the z axis of the co-rotating reference frame at node 1. These two axes remain orthogonal during the deformations.

123 M: 3d 2-node revolute joint element with 6 dof/node. This element has a co-rotating reference frame attached to each of its two nodes. Both frames have the same orientation in the underformed configuration and therefore can be viewed as far as input is concerned as the same frame. The orientation of this frame in the undeformed configuration must be specified under EFRAMES, using the same format as for a beam element. Its x axis defines the orientation in the undeformed configuration of the axis of free relative rotation.

124 M: 3d 2-node cylindrical joint element with 6 dof/node. This element has a co-rotating reference frame attached to each of its two nodes. Both frames have the same orientation in the underformed configuration and therefore can be viewed as far as input is concerned as the same frame. The orientation of this frame in the undeformed configuration must be specified under EFRAMES, using the same format as for a beam element. Its x axis defines the orientation in the undeformed configuration of the axis of free relative translation and rotation.

125 M: 3d 2-node prismatic joint element with 6 dof/node. This element has a co-rotating reference frame attached to each of its two nodes. Both frames have the same orientation in the underformed configuration and therefore can be viewed as far as input is concerned as the same frame. The orientation of this frame in the undeformed configuration must be specified under EFRAMES, using the same format as for a beam element. Its x axis defines the orientation in the undeformed configuration of the axis of free relative translation.

126 M: 3d 2-node actuated revolute joint element with 6 dof/node and a prescribed relative rotation (between node 2 and node 1) around the joint axis of rotation given by $\Delta\theta = \Delta\theta_0 \sin\omega t$. The amplitude $\Delta\theta_0$ and circular frequency ω are specified in MATERIAL. This element has a co-rotating reference frame attached to each of its two nodes. Both frames have the same orientation in the underformed configuration and therefore can be viewed as far as input is concerned as the same frame. The orientation of this frame in the undeformed configuration must be specified under EFRAMES, using the same format as for a beam element. Its x axis defines the orientation in the undeformed configuration of the axis of forced relative rotation.

127 M: 3d 2-node actuated primsatic joint element (or prismatic actuator) with 6 dof/node and a prescribed relative translation (between node 2 and node 1) along the joint axis of translation given by $\Delta u = \Delta u_0 \sin\omega t$. The amplitude Δu_0 and circular frequency ω are specified in MATERIAL.

This element has a co-rotating reference frame attached to each of its two nodes. Both frames have the same orientation in the underformed configuration and therefore can be viewed as far as input is concerned as the same frame. The orientation of this frame in the undeformed configuration must be specified under EFRAMES, using the same format as for a beam element. Its x axis defines the orientation in the undeformed configuration of the axis of forced relative translation.

201 M: 3d materially nonlinear 8-node brick element with 3 dof/node.

202 M: 3d geometrically (Green-Lagrange) + materially nonlinear 8-node brick element with 3 dof/node.

301 F: 2d 4-node sloshing (fluid) quadrilateral element with 1 dof/node.

302 F: 2d 2-node free-surface (fluid) element with 1 dof/node for two-dimensional sloshing computations using element type = 301.

311 F: 3d 4-node sloshing (fluid) tetrahedron element with 1 dof/node.

312 F: 3d 3-node free-surface (fluid) triangular element with 1 dof/node for three-dimensional sloshing computations using element type = 311.

321	F: 2d 4-node hydroelastic vibration (fluid) quadrilateral element with 1 dof/node.
331	F: 3d 4-node hydroelastic vibration (fluid) tetrahedral element with 1 dof/node.
1100	A: 2d 4-node Helmholtz DGM element Q-4-1.
1101	A: 2d 4-node Helmholtz DGM element Q-8-2.
1102	A: 2d 4-node Helmholtz DGM element Q-16-4.
1103	A: 2d 4-node Helmholtz DGM element Q-32-8.
1104	A: 2d 4-node Helmholtz DGM element Q-8-2star with additional enrichment by evanescent wave functions XXX NOT READY YET XXX.
1110	A: 2d 3-node Helmholtz DGM element T-4-1.
1111	A: 2d 3-node Helmholtz DGM element T-8-2.
1120	A: 2d 4-node Helmholtz DEM element Q-4-1.
1121	A: 2d 4-node Helmholtz DEM element Q-8-2.
1122	A: 2d 4-node Helmholtz DEM element Q-16-4.
1123	A: 2d 4-node Helmholtz DEM element Q-32-8.
1130	A: 2d 3-node Helmholtz DEM element T-4-1.
1131	A: 2d 3-node Helmholtz DEM element T-8-2.
1150	A: 3d 8-node Helmholtz DGM element H-6-1.
1151	A: 3d 8-node Helmholtz DGM element H-26-4.
1152	A: 3d 8-node Helmholtz DGM element H-56-8.
1153	A: 3d 8-node Helmholtz DGM element H-98-12.
1160	A: 2d 3-node Helmholtz DGM element T-6-1.
1161	A: 2d 3-node Helmholtz DGM element T-26-4.
1162	A: 2d 3-node Helmholtz DGM element T-56-8.
1170	A: 3d 8-node Helmholtz DEM element H-6-1.
1171	A: 3d 8-node Helmholtz DEM element H-26-4.
1172	A: 3d 8-node Helmholtz DEM element H-56-8.
1173	A: 3d 8-node Helmholtz DEM element H-98-12.
1200	M: 2d 4-node Elastodynamic DGM element Q-4x2-2.
1201	M: 2d 4-node Elastodynamic DGM element Q-16x2-8.
1220	M: 2d 4-node Elastodynamic DEM element Q-4x2-2.
1250	M: 3d 4-node Elastodynamic DGM element H-6x3-3.
1251	M: 3d 4-node Elastodynamic DGM element H-26x3-15.
1252	M: 3d 4-node Elastodynamic DGM element H-50x3-28.
	These should be listed in a stacked fashion on a single line.
CONNECTIVITY_NODES	

Next: USDD, Previous: TOPOLOGY

97 USER DEFINED FORCES *S*

Command Statement:	**USDF**

The purpose of the USDF command is to specify nodal forces via a user-defined subroutine. In this case, the user should: 1) write his/her own algorithm for specifying the nodal forces within a subroutines named “control.C”, 2) compile this subroutine and link it using the makefile provided for this command, and 3) use the LOAD command (see LOAD) to activate it. An example input file using the USDF command can be found in FEM.d/fem_examples/USDF.d/.

User-defined nodal forces are not assembled neither at the element level nor at the subdomain level. They can be combined with those specified under the FORC and GRAV command statments but will not be amplified by the MFTT or HFTT tables. In case of a conflict between this command and the FORC and GRAV commands, the specified nodal forces are simply added. A more detailed description of the “control.C” subroutine is given in its template located in the Control.d directory of **FEM**.

The syntax for invoking this option is given below.

Note 1: Similarly, this command can be used to specify time-variant prescribed Neumann boundary conditions (or source terms) for a time-domain acoustic simulation.

```
USDF

NODE#          DOF#
```

NODE#	Node number where the force is specified (integer).
DOF#	Degree of freedom local number where the force is specified (integer).

Note 1: All forces must be specified in the computational basis.

Note 2: The USDF command is redundant with the ACTUATORS command (see ACTUATORS) in the sense that ACTUATORS can achieve whatever USDF can achieve. However, the reverse is not true.

Next: WEIGHTS, Previous: USDF

98 USER DEFINED PRESCRIBED DISPLACEMENTS *S*

Command Statement:	**USDD**

The purpose of the USDD command is to specify time-variant prescribed displacements for any node and its associated dofs using a user-defined subroutine. In this case, the user should: 1) write his/her own algorithm for specifying the prescribed displacement field within a subroutine named “control.C”, 2) compile this subroutine and link it with **FEM** using the makefile that is provided for this purpose, and 3) use the LOAD command (see LOAD) to activate it. An example input file using the USDD command can be found in FEM.d/fem_examples/USDD.d/.

This command can be used simultaneously with the time-invariant command DISP for prescribing displacements at a node. In the event of a conflict between the DISP and USDD commands, USDD prevails. A more detailed description of the “control.C” subroutine is given in its template located in the Control.d directory of **FEM**.

The syntax for invoking this option is given below.

Note 1: Similarly, this command can be used to specify time-variant prescribed Dirichlet boundary conditions for a time-domain acoustic simulation by setting DOF# to 8 (see below).

```
USDD
```

```
NODE#          DOF#
```

NODE#	Node number where the displacement is specified (integer).
DOF#	Degree of freedom local number where the displacement is specified (integer).

Next: YMTT, Previous: USDD

99 WEIGHTS

Command Statement:	**WEIGHTS**

The WEIGHTS command statement can be used to modify some or all of the weights attributed by default to an element type (see TOPOLOGY). These weights are exploited by the mesh partitioning algorithm (see DECOMPOSE) to achieve load balance when generating the subdomains.

Note 1: Variable weights attributed by default to elements with a variable number of nodes cannot be modified by this command.

The input format of this command is given below.

```
WEIGHTS
```

```
ELEMENT_TYPE#        DESIRED_WEIGHT
```

ELEMENT_TYPE#	This is the element type id number as in the TOPOLOGY command (integer).
DESIRED_WEIGHT	This is the desired weight to be attributed to this element type (integer).

The weight default values are as follows:
M=Mechanics H=Thermal C=Coupled Thermoelastic F=Fluid A=Acoustic W = default weight

1	M: 3d truss (bar) element with 3 dof/node (**W = 1**).
2	M: 2d 4-node quad element with 2 dof/node (**W = 2**).
3	H: 3d 4-node quad element with 1 dof/node (**W = 2**).
4	M: 2d 3-node triangular element with 2 dof/node (**W = 2**).
6	M: 3d Bernoulli beam element with 6 dof/node (**W = 1**).
7	M: 3d Timoshenko beam element with 6 dof/node (**W = 1**).
8	M: 3d 3-AQR shell element with 6 dof/node (**W = 3**).
88	M: 3d 4-node shell element element with 6 dof/node (splits into 2 elements of type 8) (**W = 4**).
9	H: 3d 2-node lineal element with 1 dof/node (**W = 1**).
10	H: 2d 4-node quad element with 1 dof/node (**W = 2**).
11	M: 3d torsion spring element with 3 dof/node (**W = 1**).
12	C: 3d thermomelastic truss element with 4 dof/node (**W = 1**).
13	C: 2d thermoelastic quad element with 3 dof/node (**W = 2**).
14	C: 3d thermoelastic Bernoulli beam element with 7 dof/node (**W = 1**).
16	M: 3d Belytschko-Tsay shell element with 6 dof/node (**W = 3**).
17	M: 3d 8-node brick element with 3 dof/node (**W = 3**).
18*	M: 3d 4-node shear panel element with 3 dof/node (**W = 1**).
19	M: 3d 3-node membrane element with 6 dof/node (**W = 3**). (only in-plane and drilling stiffnesses) (**W = 1**).
20	M: 3d 3-node composite or orthotropic shell element with 6 dof/node (**W = 3**).
2020	M: 3d 4-node composite or orthotropic shell element with 6 dof/node (splits into 2 elements of type 20) (**W = 3**).
21	M: 3d translational spring-link element with 3 dof/node (**W = 1**).
22	M: 3d torsion spring-link element with 3 dof/node (**W = 1**).
23	M: 3d 4-node tetrahedral element with 3 dof/node (**W = 3**).
24	M: 3d 6-node pentahedral element with 3 dof/node (**W = 3**).
25	M: 3d 10-node tetrahedral element with 3 dof/node (**W = 4**).
30	A: 2d 4-node quad element with 1 dof/node (**W = 1**).
31*	A: 2d 4-node quad GLS element with 1 dof/node (**W = 1**).
32*	A: 2d 8-node quad element with 1 dof/node (**W = 3**).
33*	A: 2d 4-node quad BUBBLE element with 1 dof/node (**W = 1**).
34*	A: 2d 4-node quad two-level BUBBLE element with 1 dof/node (**W = 1**).
35	A: 2d 3-node triangular element with 1 dof/node (**W = 1**).
36	A: 2d 3-node triangular GLS element with 1 dof/node (**W = 1**).
38*	A: 2d 6-node triangular element with 1 dof/node (**W = 3**).

40	A: 3d 4-node tetrahedral element with 1 dof/node (**W = 2**).
41*	A: 3d 4-node tetrahedral GLS element with 1 dof/node (**W = 2**).
42	A: 3d 10-node tetrahedral element with 1 dof/node (**W = 3**).
44	A: 3d 8-node brick GLS element with 1 dof/node (**W = 3**).
45	A: 3d 8-node brick element with 1 dof/node (**W = 3**).
46	H: 3d 3-node triangular heat element (**W = 2**).
4646	H: 3d 4-node triangular heat element (splits into 2 elements of type 46) (**W = 3**).
47	H: 3d 2-node lineal element with 1 dof/node for boundary convection (**W = 1**).
48	H: 3d 4-node quad element with 1 dof/node for boundary convection (**W = 2**).
49	H: 3d 3-node triangular element with 1 dof/node for boundary convection (**W = 2**).
50	H: 3d 4-node tetrahedral element with 1 dof/node (**W = 3**).
51	H: 3d 8-node brick element with 1 dof/node (**W = 4**).
52*	M: 3d 6-node triangular shell element (**W = 1**).
53	H: 2d 3-node triangular heat element (**W = 2**).
56	H: 3d 2-node heat radiation element (**W = 1**).
57	H: 3d 3-node triangular heat radiation element (**W = 2**).
58	H: 3d 4-node quadrilateral heat radiation element (**W = 2**).
65	M: 3d 2-node rigid truss (bar) element (**W = 1**).
66	M: 3d 2-node rigid beam element (**W = 1**).
67	M: 3d rigid link (translational and rotational) (**W = 1**).
68	M: 3d rigid translational link (**W = 1**).
69	M: 3d rigid rotational link (**W = 1**).
70	M: 3d rigid plane or solid element with 3 dof/node (3-node to 20-node element) (**W = 1**).
71	M: 3d rigid plane or solid element with 3 dof/node and anywhere from 3 to 20 nodes per element (**W = 1**).
72	M: 3d 20-node brick element (**W = 4**).
73	M: 3d 3-node rigid shell element (**W = 1**).
74	M: 3d rigid plane or solid element with 6 dof/node (3-node to 32-node element) (**W = 1**).
81	H: 2d 4-node contact resistance thermal element with 1 dof/node that can be inserted between two thermal elements of type = 10 (**W = 2**).
82	H: 3d 8-node contact resistance thermal element with 1 dof/node that can be inserted between two thermal elements of type = 51 (**W = 4**).
83	H: 3d 6-node contact resistance thermal element with 1 dof/node that can be inserted between two thermal elements of type = 50 with 4 nodes each (**W = 4**).
84	H: 2d 3-node triangular bulk fluid (thermal) element with 1 dof/node. The first node appearing in the connectivity list of this element must be inside the bulk fluid and the other two nodes must be located at the interface boundary between the bulk fluid and the surrounding material. The temperature at the first node is by definition the average temperature of the bulk fluid (**W = 2**).
85	H: 3d 4-node tetrahedral bulk fluid (thermal) element with 1 dof/node. The first node appearing in the connectivity list of this element must be inside the bulk fluid and the other three nodes must be located at the

interface boundary between the bulk fluid and the surrounding material. The temperature at the first node is by definition the average temperature of the bulk fluid (**W = 3**).

86 H: 3d 5-node pyramidal bulk fluid (thermal) element with 1 dof/node. The first node appearing in the connectivity list of this element must be inside the bulk fluid and the other four nodes must be located at the interface boundary between the bulk fluid and the surrounding material. The temperature at the first node is by definition the average temperature of the bulk fluid (**W = 4**).

90 A: 3d 6-node wedge element with 1 dof/node (**W = 3**).

91 M: 3d 32-node serendipity brick element with 3 dof/node (**W = 6**).

92 M: 3d 26-node serendipity wedge element with 3 dof/node (**W = 5**).

93 A: 3d 32-node serendipity brick element with 1 dof/node (**W = 5**).

94 A: 3d 26-node serendipity wedge element with 1 dof/node (**W = 4**).

95 A: 3d 8-node, or 27-node, or 64-node, or 125-node hexahedral element with 1 dof/node (**W = 3, 4, 5, 6**).

96 A: 3d 4-node, or 10-node, or 20-node, or 35-node tetrahedral element with 1 dof/node (**W = 1, 2, 3, 4**).

97 M: 3d 15-node serendipity wedge element with 3 dof/node (**W = 4**).

98 A: 2d 4-node, or 9-node, or 16-node, or 25-node quadrilateral element with 1 dof/node (**W = 2, 3, 4, 5**).

99 A: 2d 3-node, or 6-node, or 10-node triangular element with 1 dof/node (**W = 2, 3, 4**).

100 M: 2d 4-node, or 9-node, or 16-node, or 25-node quadrilateral element with 3 dof/node (**W = 2, 3, 4, 5**).

101 M: 2d 3-node, or 6-node, or 10-node triangular element with 3 dof/node (**W = 2, 3, 4**).

102 M: 3d 8-node, or 27-node, or 64-node, or 125-node hexahedral element with 3 dof/node (**W = 2, 3, 4, 5**).

103 M: 3d 4-node, or 10-node, or 20-node, or 35-node tetrahedral element with 3 dof/node (**W = 1, 2, 3, 4**).

105 A: 3d 8-node, or 27-node, or 64-node, or 125-node spectral hexahedral element with 1 dof/node (**W = 2, 3, 4, 5**).

108 A: 2d 4-node, or 9-node, or 16-node, or 25-node spectral quadrilateral element with 1 dof/node (**W = 2, 3, 4, 5**).

111 M: 3d fabric truss element with 3 dof/node (**W = 1**).

120 M: 3d 2-node spherical joint element with 3 dof/node (**W = 1**).

122 M: 3d 2-node universal joint element with 6 dof/node (**W = 1**).

123 M: 3d 2-node revolute joint element with 6 dof/node (**W = 1**).

124 M: 3d 2-node cylindrical joint element with 6 dof/node (**W = 1**).

125 M: 3d 2-node prismatic joint element with 6 dof/node (**W = 1**).

126 M: 3d 2-node actuated revolute joint element with 6 dof/node (**W = 1**).

127 M: 3d 2-node actuated prismatic joint element with 6 dof/node (**W = 1**).

201 M: 3d materially nonlinear 8-node brick element with 3 dof/node (**W = 3**).

202 M: 3d geometrically (Green-Lagrange) + materially nonlinear 8-node brick element with 3 dof/node (**W = 3**).

301 F: 2d 4-node sloshing (fluid) quadrilateral element with 1 dof/node (**W = 1**).

302 F: 2d 2-node free-surface (fluid) element with 1 dof/node for two-dimensional sloshing computations using element type = 301 (**W = 1**).

311 F: 3d 4-node sloshing (fluid) tetrahedron element with 1 dof/node (**W = 3**).

312 F: 3d 3-node free-surface (fluid) triangular element with 1 dof/node for three-dimensional sloshing computations using element type = 311 (**W = 2**).

321 F: 2d 4-node hydroelastic vibration (fluid) quadrilateral element with 1 dof/node (**W = 2**).

331 F: 3d 4-node hydroelastic vibration (fluid) tetrahedral element with 1 dof/node (**W = 3**).

1100 A: 2d 4-node Helmholtz DGM element Q-4-1 (**W = 1**).

1101 A: 2d 4-node Helmholtz DGM element Q-8-2 (**W = 2**).

1102 A: 2d 4-node Helmholtz DGM element Q-16-4 (**W = 4**).

1103 A: 2d 4-node Helmholtz DGM element Q-32-8 (**W = 8**).

1104 A: 2d 4-node Helmholtz DGM element Q-8-2star with additional enrichment by evanescent wave functions XXX NOT READY YET XXX (**W = 2**).

1110 A: 2d 3-node Helmholtz DGM element T-4-1 (**W = 1**).

1111 A: 2d 3-node Helmholtz DGM element T-8-2 (**W = 2**).

1120 A: 2d 4-node Helmholtz DEM element Q-4-1 (**W = 1**).

1121 A: 2d 4-node Helmholtz DEM element Q-8-2 (**W = 2**).

1122 A: 2d 4-node Helmholtz DEM element Q-16-4 (**W = 4**).

1123 A: 2d 4-node Helmholtz DEM element Q-32-8 (**W = 8**).

1130 A: 2d 3-node Helmholtz DEM element T-4-1 (**W = 1**).

1131 A: 2d 3-node Helmholtz DEM element T-8-2 (**W = 2**).

1150 A: 3d 8-node Helmholtz DGM element H-6-1 (**W = 1**).

1151 A: 3d 8-node Helmholtz DGM element H-26-4 (**W = 4**).

1152 A: 3d 8-node Helmholtz DGM element H-56-8 (**W = 8**).

1153 A: 3d 8-node Helmholtz DGM element H-98-12 (**W = 12**).

1160 A: 2d 3-node Helmholtz DGM element T-6-1 (**W = 1**).

1161 A: 2d 3-node Helmholtz DGM element T-26-4 (**W = 4**).

1162 A: 2d 3-node Helmholtz DGM element T-56-8 (**W = 8**).

1170 A: 3d 8-node Helmholtz DEM element H-6-1 (**W = 1**).

1171 A: 3d 8-node Helmholtz DEM element H-26-4 (**W = 4**).

1172 A: 3d 8-node Helmholtz DEM element H-56-8 (**W = 8**).

1173 A: 3d 8-node Helmholtz DEM element H-98-12 (**W = 12**).

1200 M: 2d 4-node Elastodynamic DGM element Q-4x2-2 (**W = 2**).

1201 M: 2d 4-node Elastodynamic DGM element Q-16x2-8 (**W = 8**).

1220 M: 2d 4-node Elastodynamic DEM element Q-4x2-2 (**W = 2**).

1250 M: 3d 4-node Elastodynamic DGM element H-6x3-3 (**W = 3**).

1251 M: 3d 4-node Elastodynamic DGM element H-26x3-15 (**W = 15**).

1252 M: 3d 4-node Elastodynamic DGM element H-50x3-28 (**W = 28**).

Previous: WEIGHTS

100 YOUNGS MODULUS TEMPERATURE TABLE

Command Statement:	**YMTT**

The YMTT command statement can be used to describe the evolution of Young's modulus with temperature, for a given material. This evolution can be specified here in a curve (or table) defined by pairs of temperature and Young's modulus values. Linear interpolation is used for “in between” points, and the extrema values are adopted for “outside” points. Several curves can be specified, one after the other. Each curve is identified by an “id number” as described below.

```
YMTT
```

```
CURVE curve_id
T_1   YM_1
.
.
.
T_1   YM_1
```

CURVE	
curve_id	“Id number” for the following curve (or table) (integer).
T_1	A specified temperature value (float).
YM_1	A specified Young's modulus value at temperature T_1 (float).