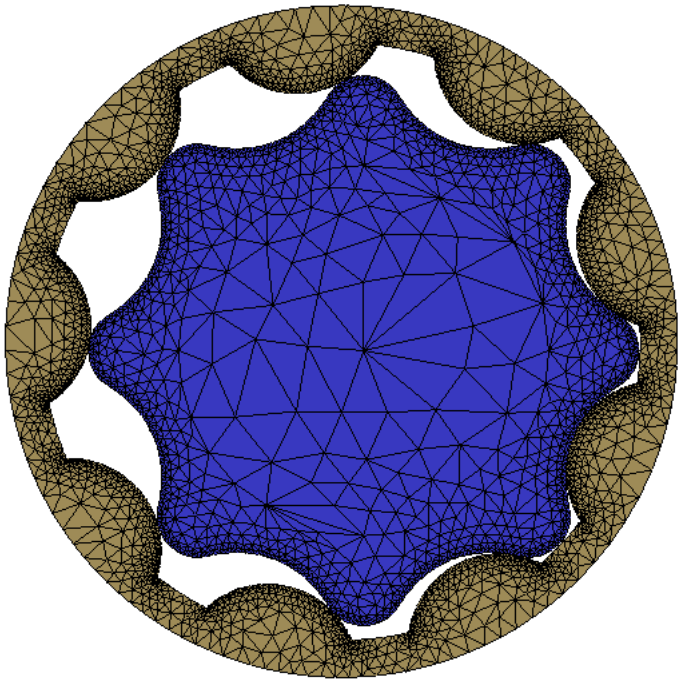
COMET

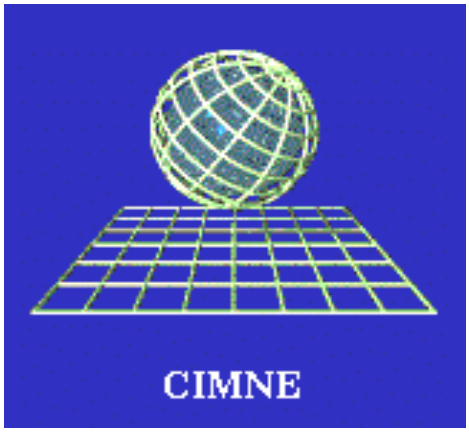
Coupled Mechanical and Thermal Analysis



DATA INPUT MANUAL

Version 5.0

International Center for Numerical Method in Engineering



Release June 2002

M. Cervera, C. Agelet de Saracibar and M. Chiumenti

INDEX

1. INTRODUCTION

COMET is a program developed at the International Center for Numerical Methods in Engineering of Barcelona (CIMNE). Its name means COupled MEchanical and Thermal problems solved by the Finite Element Method (FEM).

COMET is a modular program intended to encompass a wide range of applications, and to be used as a practical tool both in industrial and research environments.

COMET is a program that may perform the following Finite Element Analyses:

⇒ **Mechanical** 2D or 3D problems subjected to static, quasi-static or transient dynamic conditions.

⇒ **Thermal** 2D or 3D problems subjected to transient or steady state conditions.

⇒ **Thermal-Mechanical** 2D/3D problems in transient or steady state conditions.

The solution procedure in a non linear problem may be helped by the following options:

* Initial stiffness matrix, or full Newton-Raphson, or modified Newton-Raphson type of overall algorithm;

* Line search;

* Convergence accelerators: Secant-Newton (1 or 2 parameters), and BFGS;

* Arc-length and displacement control;

Other particular features are:

- The integration may be performed by Gauss, or Lobatto, or Irons rules;
- The system of linear equations may be solved by
  - Direct solver using a Skyline storage scheme
  - Preconditioned Conjugate Gradient Iterative solver
  - GMRES Iterative solver;
- The time discretization scheme may be selected among the Wilson (1^{st}. order) or Newmark (2^{nd} order) schemes.

1.1 Generality of the input data structure:

All the data input for COMET is read through a *command interpreter.* **Free format** may therefore be used throughout. Other input facilities are also provided to allow the creation of meaningful and easy-to-read data input files.

It is possible to write as many comments as needed throughout the input data command sequence. A **comment line** starts with the character '$' in the first column. It is also possible to include **in-line comments** (FORTRAN style) where all text following the character '!' will be ignored on that line.

The **continuation line** symbol is '/' or '\'. Comments may also be written after the continuation symbols.

Not all the data must be contained in a unique data file. The input may be redirected to any file desired using the **INCLUDE** option. When the INCLUDE directive is found anywhere in the data file, the input procedure is immediately redirected to the specified file. Return to the main data file is automatic when all the file has been read. The line following the INCLUDE will then be read.

1.2 Data input format

The general format of all COMET commands is

Command <Qualifiers> <Parameters>

where:

Command: is the key word. Only one command is allowed on a single line; if the command is optional the defaults are used if nothing is specified.

Qualifiers: the user may select from a list of possible choices.

Parameters: their general format is:

Parameters = XX

where XX is either a real or an integer number. Sometimes they are optional; if this is the case default values will be used if nothing is specified.

1.3 Free format input

COMET allows the user to write the input datafile using a free format scheme. In this case the data is interpreted taking into account the following rules:

- Numbers, commands, qualifiers, parameters can be separated either by a comma (,) or by a colon (:) or by a blank ( ) character.
- Numbers can be written in real, integer or exponential format.
- The command interpreter is case insensitive.
- The continuation symbol for commands longer than 80 characters is either / or \.
- The maximum length of a command is 80 columns.
- Comment cards can be inserted everywhere with the character $ in first column.

  In line comments can be inserted after the character !.

List specification is allowed for initial and boundary conditions: the command words TO and SKIP (default is 1) can be used. The following are valid examples of list specification commands:

```
1  TO  9  SKIP  2     110   0.   0.   0.1

61  TO  67     21   1.   0.   0.2
```

1.4 Include commands

The use of the INCLUDE command allows the user to keep sections of the input file separate thereby making the input file clearer and more easy to understand.

The files to be included must contain the commands, parameters and qualifiers written according to the syntax rules specified in this manual and must end with the word EOF.

In the input file, after the command INCLUDE, the correct path of the file to be included has to be specified, paying attention to lower and upper case letters to which the operating system is sensitive. For example the following is a valid command when using a UNIX operating system:

```
INCLUDE     /usr/users/COMET_user/Test/exampleX.mesh
```

The contents of the include file are then effectively placed in the input datafile instead of the INCLUDE command.

1.5 Sequence of input data

The COMET input data has a defined format that allows the user maximum control over the analysis run. This allows the possibility of changing, within the same run, the applied loads, boundary conditions and the solution strategy during the analysis procedure. The input data sequence is depicted in Figures 1a and 1b and is described below.

- The input data starts with a header command COMET followed by the title of the run.

- Next the compulsory START command states if the analysis to be performed is a new run or a restart run.

After these commands the data blocks following the order specified in Figure 1a must be inserted. Each data block must begin with a specified block_data command and end with an end_block_data command which are compulsory since they give the possibility to access and to leave the input data sequence of each particular block. The order of the data blocks is related to the calculation flow, as their sequence define a procedure.

- CONTROL_DATA block defines the type of analysis to be performed. By default the program is set to perform an *uncoupled plane-strain elastic problem* of a *single-phase* medium subjected to *static loading* conditions with *small displacements*. Any changes from these default options can be made with the appropriate input commands (see the CONTROL_DATA section of this manual). The CONTROL_DATA block *must* include a DIMENSIONS command defining the maximum dimensions to be used in the problem to set the memory requirements

- GENERAL_DATA block defines the problem that is going to be solved: in this section must be defined the following blocks:

  - GEOMETRY block defines the geometry and the topology of the mesh to be analysed.

  - SETS block defines the type of element, the integration rule and the materials used in each set defined.

  - CONTACT_DATA block defines the geometry, topology, and characteristics of the contact surfaces.

  - INITIAL_DATA block defines the initial conditions such as initial temperature, displacements and velocity fields. By default the run starts with zero initial conditions, that is zero initial state variables. If non-zero initial conditions are required the appropriate data must be provided (see the INITIAL_DATA section of this manual).

- INTERVAL_DATA block is itself subdivided into four blocks: it is possible to define sequentially several INTERVAL_DATA blocks have the possibility of changing the boundary conditions, or the loads applied or the solution strategy. To change the functions, the reference load or displacement, the boundary conditions or the solution strategy a new INTERVAL_DATA must be defined within which only those options which are required to change from one interval to the next need to be specified.

  - FUNCTION_DATA specifies the functions used to define the loads or the boundary time incrementation. The data in this block allows time functions to be built up for the current time interval.

  - LOAD_DATA block specifies the reference load conditions for the current time interval. This reference load may be composed of point, surface and/or volume load components. The actual loads (and imposed variables) for each time step are computed using the defined load functions (see LOAD_DATA section of this manual).

  - BOUNDARY_DATA block defines the reference fixity conditions, i.e. the Dirichlet conditions (see the BOUNDARY_DATA section of this manual).

  - STRATEGY_DATA block identifies the solution strategy to be followed for the particular time interval. By default the program performs an analysis without updating the initial stiffness matrix. Any other solution strategy must be specified via the proper input commands. Other parameters such the desired frequency for output and restart dumping must also be specified (see the STRATEGY_DATA sections of this manual).

- Input data must end with the STOP command.

COMET	*<qualifiers>*

START	*<qualifiers> <parameters=#>*

CONTROL_DATA
....
END_CONTROL_DATA

GENERAL_DATA
....
END_GENERAL_DATA

INTERVAL_DATA	First Interval
....	
END_INTERVAL_DATA	

INTERVAL_DATA	Second Interval
....	
END_INTERVAL_DATA	

....

INTERVAL_DATA	Last Interval
....	
END_INTERVAL_DATA	

STOP

```
CONTROL_DATA
    [PROBLEM DEFINITION]
END_CONTROL_DATA
```

```
GENERAL_DATA

    GEOMETRY
        [connectivities]
        [coordinates]
    END_GEOMETRY

    SETS
        [element and material definitions]
    END_SETS

    CONTACT_DATA
        [control parameters]
        [slidelines geometry]
    END_CONTACT_DATA

    INITIAL_DATA
        [initial conditions]
    END_INITIAL_DATA

END_GENERAL_DATA
```

```
INTERVAL_DATA

    FUNCTION
        [function definition]
    END_FUNCTION

    BOUNDARY
        [boundary conditions]
    END_BOUNDARY

    LOAD
        [load conditions]
    END_LOAD

    ACTIVATION
        [activation strategy]
    END_ ACTIVATION

    STRATEGY
        [strategy solution options]
    END_STRATEGY

END_INTERVAL_DATA
```

2. SPECIAL COMMANDS

These commands are not collected in a specific data block as they perform specific actions related to the external control of the run.

Their attributes define the job identifier, the type of session (either START or RESTART analysis), information for external file management and system.

2.1 COMET *<qualifiers>*

compulsory command

Description:

This command indicates the beginning of the input data file.

Qualifiers:

<TITLE> The title of the analysis as a character string.

Notes:

This command must be the first line of the data file.

Example:

```
COMET:   this is the test example number 361
```

2.2 START *<qualifiers> <parameters=#>*

compulsory command

Description:

This command declares the start of the data file.

Qualifiers:

NEW_RUN	The following data file is for a new analysis.
CONTINUE	Restart facility: the run will start from the *last record* saved in the restart file *name.*rst . Data file will be used to continue the analysis from that point.
FROM	Restart facility: the run will start from the specified INTERVAL and STEP This record will be read from the restart file *name.*rst . Data file will be used to continue the analysis from that point.

Parameters:

INTERVAL=#	Specify the INTERVAL to restart the analysis (only for restart analysis with option FROM).
STEP=#	Specify the STEP to restart the analysis (only for restart analysis with option FROM).
TIME=#	The starting-time from which the analysis will start (may be specified as non-zero. Only for NEW_RUN option).
CPU_LIMIT=#	Specification of a CPU limit (in seconds) for the run (only for NEW_RUN option).

Defaults:

The defaults are NEW_RUN and TIME=0.0

Notes:

To use the restart option it must exist the restart file called *name.rst* in the directory where the results will be dumped. It is also necessary to launch COMET with the option *[-restart]* not to remove the restart file. Note that the restart file is created when the SAVE option is used in the STRATEGY_data block when running a NEW_RUN calculation.

Examples:

```
START :   NEW_RUN  CPU_LIMIT=10E+8
```

```
START :   CONTINUE
```

```
START :  FROM   INTERVAL=3   STEP=1
```

2.3 STOP

compulsory command

Description:

This command declares the end of the data file.

- It is possible to put the STOP card at the end of the GEOMETRY_DATA block to check the input of the geometry, material and contact information before running all the analysis.
- Alternatively, the STOP card can be used to stop the analysis after any INTERVAL_DATA block.

Notes:

The STOP command must be the last card of data file. All the information given after the STOP command will not be read.

2.4 INCLUDE *<qualifiers>*

optional command

Description:

This card addresses the input procedure to the specified file. The contents of the specified file are effectively included in the main data file. Reading of input returns to the main data file automatically at the EOF.

Parameters:

<FILENAME> The full path for the file to be included.

Notes:

Use the full path name taking into account lower and upper case characters and the operating system that is being used.

Examples:

```
INCLUDE   /usr/COMET_user/data/COMET_files/example23.geo
```

3. CONTROL_DATA COMMANDS

The type and dimension of the problem to be solved in terms of basic memory requirements is defined here.

CONTROL_DATA

END_CONTROL_DATA

compulsory cards

Description:

This command cards indicate the beginning and the end of the CONTROL_data block.

Structure:

CONTROL_DATA	
GEOMETRY	<*qualifiers*>
MECHANICAL	<*qualifiers*>
THERMAL	<*qualifiers*>
COUPLED	<*qualifiers*>
DIMENSIONS	<*parameters=#*>
CONTACT	<*qualifiers*> <*parameters=#*>
ACTIVATION	<*qualifiers*>
SOLVER	<*qualifiers*> <*parameters=#*>
SUBDOMAIN	<*qualifiers*> <*parameters=#*>
POST_PROCESS	<*qualifiers*>
SMOOTHING	<*qualifiers*>
END_CONTROL_DATA	

3.1 GEOMETRY *<qualifiers>*

compulsory command

Description:

The command defines the geometrical definition of the problem.

Qualifiers:

TWODI or 2D	Two dimensional problem (XY plane-strain assumption)
AXISYMM	Axialsymmetrical problem (Y is the rotational axe)
THREEDI or 3D	Three dimensional problem

Default:

GEOMETRY: 2D

Example:

```
CONTROL_DATA
   GEOMETRY:       3D
   MECHANICAL      <qualifiers>
   DIMENSIONS      <parameters=#>
   CONTACT         <qualifiers>  <parameters=#>
   ACTIVATION      <qualifiers>
   SOLVER          <qualifiers>  <parameters=#>
   SUBDOMAIN       <qualifiers>  <parameters=#>
   POST_PROCESS    <qualifiers>
   SMOOTHING       <qualifiers>
END_CONTROL_DATA
```

3.2 MECHANICAL *<qualifiers>*

optional command

Description:

This command specifies that a MECHANICAL problem is to be solved. This option excludes the use of THERMAL and COUPLED commands.

Qualifiers:

DYNAMIC	Mechanical behaviour is DYNAMIC, inertial and damping terms are considered.
STATIC	Mechanical behaviour is QUASI-STATIC, inertial and damping terms are neglected.
MIXED	In this case the code solves the PRESSURE as a nodal variable a a part from the standard displacements. This option is only usable together with the appropriate element technology (see elements definition)

Defaults:

MECHANICAL: STATIC

Example:

```
CONTROL_DATA
   GEOMETRY       <qualifiers>
   MECHANICAL:    STATIC  MIXED
   DIMENSIONS     <parameters=#>
   CONTACT        <qualifiers>  <parameters=#>
   ACTIVATION     <qualifiers>
   SOLVER         <qualifiers>  <parameters=#>
   SUBDOMAIN      <qualifiers>  <parameters=#>
   POST_PROCESS   <qualifiers>
   SMOOTHING      <qualifiers>
END_CONTROL_DATA
```

3.3 THERMAL *<qualifiers>*

optional command

Description:

This command specifies that a thermal problem is to be solved. This option excludes the use of MECHANICAL and COUPLED commands.

Qualifiers:

TRANSIENT	Thermal behaviour is TRANSIENT, time derivative terms are considered.
STEADY_STATE	Time derivative terms are neglected.

Default:

THERMAL: TRANSIENT

Example:

```
CONTROL_DATA
   GEOMETRY       <qualifiers>
   THERMAL:       TRANSIENT
   DIMENSIONS     <parameters=#>
   CONTACT        <qualifiers>  <parameters=#>
   ACTIVATION     <qualifiers>
   SOLVER         <qualifiers>  <parameters=#>
   SUBDOMAIN      <qualifiers>  <parameters=#>
   POST_PROCESS   <qualifiers>
   SMOOTHING      <qualifiers>
END_CONTROL_DATA
```

3.4 COUPLED *<qualifiers>*

optional command

Description:

This command specifies a coupled problem (e.g. thermo-mechanical). The qualifiers determine the sequence and the type of the staggered solution. This option excludes the use of THERMAL and MECHANICAL commands.

Qualifiers:

MEC_TH	The sequence of the solution is MECHANICAL ⇒ THERMAL.
TH_MEC	The sequence of the solution is THERMAL ⇒ MECHANICAL
DYNAMIC	The mechanical partition is time dependent and behaves dynamically.
MIXED	In this case the mechanical partition solves the PRESSURE as a nodal variable a a part from the standard displacements. This option is only usable together with the appropriate element technology (see elements definition)

Notes:

It is not necessary for both the thermal and mechanical partitions to be time dependent. Often the mechanical partition does not have inertial effects and then this may be analysed quasi-statically. Option TH_MEC or MEC_TH define the order in which the two partitions will be solved. The program will look for the convergence of each partition one after the other without iterating between the to partition according to a fractional step method (see COMET theory manual).

Examples:

```
CONTROL_DATA
   GEOMETRY       <qualifiers>
   COUPLED:       TH_MEC
   DIMENSIONS     <parameters=#>
   CONTACT        <qualifiers>  <parameters=#>
   ACTIVATION     <qualifiers>
   SOLVER         <qualifiers>  <parameters=#>
   SUBDOMAIN      <qualifiers>  <parameters=#>
   POST_PROCESS   <qualifiers>
   SMOOTHING      <qualifiers>
END_CONTROL_DATA
```

3.5 DIMENSIONS *<parameters=#>*

compulsory command

Description:

The command defines the dimensions of the problem. The information given in this line will be used to allocate memory for the considered analysis: a possible mistake will generate a segmentation fault during the program execution.

Parameters:

NPOIN=#	Number of nodal points (compulsory)
NELEM=#	Number of elements (compulsory)
NNODE=#	Maximum number of nodes per element (compulsory)
NGAUS=#	Maximum number of integration points per element (compulsory)
NSETS=#	Number of sets, (default = 1).
MCURV=#	Maximum number of load-functions curves, (default = 1).

Examples

```
CONTROL_DATA
   GEOMETRY       <qualifiers>
   MECHANICAL     <qualifiers>
   DIMENSION      NPOIN= 20  NELEM = 30   NNODE = 8  NGAUS = 4  /
                  NSETS = 3  MCURV = 2
   CONTACT        <qualifiers>  <parameters=#>
   ACTIVATION     <qualifiers>
   SOLVER         <qualifiers>  <parameters=#>
   SUBDOMAIN      <qualifiers>  <parameters=#>
   POST_PROCESS   <qualifiers>
   SMOOTHING      <qualifiers>
END_CONTROL_DATA
```

3.6 CONTACT *<qualifier> <parameters=#>*

optional command

Description:

This command specifies that the problem to be analysed involves mechanical, thermal or thermo-mechanical contact. This command line is used for the memory allocation of the contact database and for automatic generation of the contact surfaces (if required)

Qualifiers (Small displacements formulation):

GENERATE Using this option it is possible to generate contact surfaces according to the user requirements. Contact surfaces will be generated at the interface between different sets (see contact data block)

Note:

If contact generation is required the number of nodes and elements will be modified allowing the duplication of all the nodes at the interface between the pairs of sets where the contact is specified in the contact data block definition. Contact-joint face-to-face elements will be generated (see elements definition).

Example:

```
CONTROL_DATA
   GEOMETRY       <qualifiers>
   MECHANICAL     <qualifiers>
   DIMENSIONS     <parameters=#>
   CONTACT:         GENERATE
   ACTIVATION     <qualifiers>
   SOLVER         <qualifiers>  <parameters=#>
   SUBDOMAIN      <qualifiers>  <parameters=#>
   POST_PROCESS   <qualifiers>
   SMOOTHING      <qualifiers>
END_CONTROL_DATA
```

Qualifiers (Large displacements formulation):

SINGLE	There exists at least one single surface.
TWO_PASS	The TWO_PASS contact algorithm is required at least in one of the contact slideline definition in the CONTACT_DATA block.
GAUSSIAN	GAUSSIAN quadrature is used in the contact algorithm to integrate at least one contact slideline.

Parameters (Large displacements formulation):

NSLID=#	Total number of slidelines.
NNPST=#	Total number of nodal points in all slave surfaces (only 2D or AXIAL).
NNPMT=#	Total number of nodal points in all master surfaces (only 2D or AXIAL).
NSFTL=#	Total number of facets in all slave surfaces (only 3D).
NMFTL=#	Total number of facets in all master surfaces (only 3D).
MNSF=#	Maximum number of nodes per slave facet in any slideline.
MNMF=#	Maximum number of nodes per master facet in any slideline.

Example:

```
CONTROL_DATA
   GEOMETRY       <qualifiers>
   MECHANICAL     <qualifiers>
   DIMENSIONS     <parameters=#>
   CONTACT:        NSLID=1  NSFTL=100   NMFTL=200  MNSF=3   MNMF=3
   DATABASE       <qualifiers>
   ACTIVATION     <qualifiers>
   SOLVER         <qualifiers>  <parameters=#>
   SUBDOMAIN      <qualifiers>  <parameters=#>
   POST_PROCESS   <qualifiers>
   SMOOTHING      <qualifiers>
END_CONTROL_DATA
```

3.7 ACTIVATION *<qualifier>*

optional command

Description:

This command specifies that the problem to be analysed involves the activation/deactivation procedure. The following qualifiers permits to initialise the status of the elements/sets and contact slidelines defined.

Qualifiers:

ON	All the elements and slidelines declared are activated.
OFF	All the elements and slidelines declared will be not activated.

Default:

ACTIVATION: ON

Example:

```
CONTROL_DATA
   GEOMETRY       <qualifiers>
   MECHANICAL     <qualifiers>
   DIMENSIONS     <parameters=#>
   CONTACT        <qualifiers><parameters=#>
   ACTIVATION:    ON
   SUBDOMAIN      <qualifiers>  <parameters=#>
   SOLVER         <qualifiers>  <parameters=#>
   POST_PROCESS   <qualifiers>
   SMOOTHING      <qualifiers>
END_CONTROL_DATA
```

3.8 SOLVER *<qualifiers> <parameters>*

optional command

Description:

This command specifies the system equation solver (linear set of equations) to be used in the analysis.

Qualifiers:

DIRECT — Direct solver using a SKYLINE storage will be actived.

- RENUMBER — Internal node renumbering
- NO_RENUMBER — Suppress internal node renumbering.

ITERATIVE — Iterative solver will be used The type of solver as welle as the type of preconditioner can be defined by the user according to the following options:

a) Type of solver:

- CGRAD — Conjugate gradient
- BICON — BI-Conjugate gradient.
- GMRES — GMRES.

b) Type of preconditioner:

- DIAGONAL — Diagonal matrix pre-conditioning
- LEFT — Left matrix pre-conditioning.
- RIGHT — Right matrix pre-conditioning.
- TOTAL — Global matrix pre-conditioning

SYMMETRIC — Symmetric system of equations for all the partitions

UNSYMMETRIC — Non-symmetric system of equations for all the partitions

UNSYMMETRIC=# — Non-symmetric system of equations for partition number #.

Parameters:

MAXIT=# — Maximum number of iterations admitted for ITERATIVE solvers.

TOLER=# — Convergence TOLERANCE for ITERATIVE solvers based on the norm of the right hand side of the system of equations. Default: TOLER=1.0E-6 .

KRYLO=# — Dimension of the Krylov Subspace for the ITERATIVE UNSYMMMETRIC solver (GMRES). Default: KRYLO=20.

Default:

- SOLVER: DIRECT

  SYMMETRIC RENUMBER options will be considered as default option.
- SOLVER: ITERATIVE

  SYMMETRIC CGRAD DIAGONAL options will be considered as default option.
- SOLVER: ITERATIVE UNSYMMETRIC

  GMRES DIAGONAL options will be considered as default option.

Notes:

- This command can be alternatively defined in the STRATEGY_DATA block.
- The RENUMBER option can be used if it exists a physical connection between all the continuum body involved in the analysis: the connection could be achieved through the definition of contact slidelines.
- For coupled analysis with two partitions (mechanical and thermal) the UNSYMMETRIC qualifier may be specified for a particular partition.

  Example:

UNSYMMETRIC	means both partitions are unsymmetric;
UNSYMMETRIC=1	means only the first partition is unsymmetric;
UNSYMMETRIC=2	means only the second partition is unsymmetric;

- The qualifiers are grouped above to show their exclusivity. If combinations of qualifiers from a group are used the last one will always be utilised. Some qualifiers in certain combinations may be ignored. See the examples below.

Example:

```
CONTROL_DATA
   GEOMETRY       <qualifiers>
   MECHANICAL     <qualifiers>
   DIMENSIONS     <parameters=#>
   CONTACT        <qualifiers>  <parameters=#>
   ACTIVATION     <qualifiers>
   SOLVER         DIRECT
   SUBDOMAIN      <qualifiers>  <parameters=#>
   POST_PROCESS   <qualifiers>
   SMOOTHING      <qualifiers>
END_CONTROL_DATA
```

```
CONTROL_DATA
   GEOMETRY       <qualifiers>
   MECHANICAL     <qualifiers>
   DIMENSIONS     <parameters=#>
   CONTACT        <qualifiers>  <parameters=#>
   ACTIVATION     <qualifiers>
   SOLVER         ITERATIVE   SYMMETRIC   MAXIT=10000   TOLER=1.0E-6
   SUBDOMAIN      <qualifiers>  <parameters=#>
   POST_PROCESS   <qualifiers>
```

```
    SMOOTHING        <qualifiers>
END_CONTROL_DATA
```

3.9 SUBDOMAIN *<qualifier> <parameters=#>*

optional command

Description:

This command specifies that the problem to be analysed involves subdomain decomposition. This methodology can be employed in case of MECHANICAL or THERMO-MECHANICAL analysis including MECHANICAL-CONTACT to overcome the problem of ill-conditioning that is intrinsic in the contact algorithm when PENALTY formulation is used.

Qualifiers:

CONTACT	The equations to be solved will be spited into two blocks: the first block will recollect all the equation that include contact contributions while the second one will consider all the other equations. A block-iterative algorithm will be used to reach the convergence between the solution of these blocks.
SETS	This qualifier can be used only if CONTACT qualifier is used. It specify another a domain decomposition according to the definition of SETS.

Parameters:

TOLER=#	Specify the tolerance to be used in the block-iterative algorithm that deals with the domain decomposition.
MAXIT=#	Specify the maximum number of iteration to be performed to reach the convergence within the block-iterative algorithm that deals with the domain decomposition.

Example:

```
CONTROL_DATA
   GEOMETRY       <qualifiers>
   MECHANICAL     <qualifiers>
   DIMENSIONS     <parameters=#>
   CONTACT        <qualifiers>  <parameters=#>
   ACTIVATION     <qualifiers>
   SUBDOMAIN:        CONTACT    TOLER=1.0   MAXIT=10
   SUBDOMAIN      <qualifiers>  <parameters=#>
   POST_PROCESS   <qualifiers>
   SMOOTHING      <qualifiers>
END_CONTROL_DATA
```

3.10 POST_PROCESS *<qualifiers>*

optional command

Description:

This card indicates that a post-processor is used after the COMET run. A post-process file will be opened and results will be written into it. The dumping frequency of the results to this file is defined in each interval in the STRATEGY_DATA block.

Qualifiers:

FORMATTED	Formatted post-process file (compatible with GID input format)

Default:

The default is the FORMATTED post-process file.

Note:

In case of the post-process of gaussian variables is necessary to define the SMOOTHING card.

Example:

```
CONTROL_DATA
   GEOMETRY       <qualifiers>
   MECHANICAL     <qualifiers>
   DIMENSIONS     <parameters=#>
   CONTACT        <qualifiers>  <parameters=#>
   ACTIVATION     <qualifiers>
   SOLVER         <qualifiers>  <parameters=#>
   SUBDOMAIN      <qualifiers>  <parameters=#>
   POST_PROCESS:  FORMATTED
   SMOOTHING      <qualifiers>
END_CONTROL_DATA
```

3.11 SMOOTHING *<qualifiers>*

optional command

Description:

This command indicates that smoothing for the stresses and principal must be performed to produce a continuous nodal field. It is possible to include smoothing for internal variables defined at gauss points.

Qualifiers:

NONE	Smoothing procedure is not required.
DIRECT	Global smoothing (direct lumping or Fusco method) to be used.
HINTON	Global smoothing (Hinton and Campbell lumping) to be used.
DISCRETE	Discrete smoothing to be used.
LOCAL	Local smoothing to be used.

Default:

The default is the DIRECT smoothing procedure.

Notes:

LOCAL smoothing cannot be used with triangular elements.

Examples:

```
CONTROL_DATA
    MECHANICAL      <qualifiers>
    DIMENSIONS      <parameters=#>
    CONTACT         <qualifiers>  <parameters=#>
    ACTIVATION      <qualifiers>
    SOLVER          <qualifiers>  <parameters=#>
    SUBDOMAIN       <qualifiers>  <parameters=#>
    POST_PROCESS    <qualifiers>
    SMOOTHING:      DIRECT
END_CONTROL_DATA
```

4. GENERAL_DATA COMMANDS

Geometry sets and material properties data are supplied according to prescribed formats. Typically this block is prepared by means of a pre-processor and then included via INCLUDE command.

GENERAL_DATA

END_GENERAL_DATA

compulsory cards

Description:

This command cards indicate the beginning and the end of the GENERAL_data block.

Structure:

```
GENERAL_DATA
    GEOMETRY
        [connectivities]
        [coordinates]
    END_GEOMETRY
    SETS
        [element and material definitions]
    END_SETS
    CONTACT_DATA
        [control parameters]
        [slidelines geometry]
    END_CONTACT_DATA
    INITIAL_DATA
        [initial conditions]
    END_INITIAL_DATA
END_GENERAL_DATA
```

4.1 GEOMETRY *<qualifiers> <parameters=#>*

compulsory command

Description:

This card activates the input procedure for the geometry data. The qualifier specifies if the program should compute the coordinates of the mid-side nodes automatically. In this case the mid-node coordinates are computed if, and only if, they are zero. If the geometry should be read from the data file, the geometry data cards must immediately follow this card.

Qualifiers:

INTERPOLATE	The program automatically computes the coordinates of the mid-nodes if they are not specified. They are calculated as a linear interpolation of the corner point node coordinates;
NO_INTERPOLATE	The program does not perform any computation on the coordinates of any mid side node.
SISMM	The mesh coordinates will be divided by 1000. With this option is so possible to input the mesh coordinates in [mm] and the material properties in SI.

Parameters:

SCALE=#	The mesh coordinates will be multiplied by SCALE.

Default:

GEOMETRY: NO_INTERPOLATE

Structure:

The geometry data block is made up of 2 datasets:

(1) CONNECTIVITY set where 1 line is input for each element;

(2) COORDINATES set where 1 line is input for each node.

The structure of these datasets is detailed below.

GEOMETRY	
[CONNECTIVITY]	*<elements_connectivity>*
[COORDINATES]	<nodal_coordinates>
END_GEOMETRY	

[CONNECTIVITY] *<element connectivity>*

compulsory line, one line for each element

ELEM=#	Element number.
SET=#	Set identification number for the element.
NODES=<nodal_list>	Element connectivity list

[COORDINATES] *<nodal_coordinates>*

compulsory line, one line for each node

NODE=#	Nodal point number.
X=# Y=# Z=#	x (or r), y (or z), z coordinate of the node.

Notes:

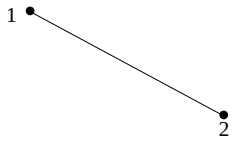
- If the qualifier INTERPOLATE is specified then the total number of cards in the Dataset 2 will differ from the total number of nodes (mid-nodes will be interpolated by the program)..
- For Lagrangian elements (i.e. the 9 node quadrilateral, the 10 node tetrahedral and the 27 node hexahedral elements, see Figure 2) the coordinates of the central node are never input. The total number of cards in the Dataset 2 will also differ from the total number of nodes.
- This data block must end with the card END_GEOMETRY

Examples:

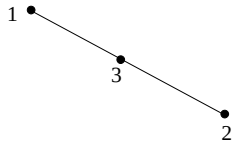
```
GENERAL_DATA
   GEOMETRY  INTERPOLATE  !  Parabolic 2D quadrilateral element with 9 nodes
      1  1      1  3  5  7  2  4  6  8  9
      1  0.0    0.0
      3  1.0    0.0
      5  1.0    1.0
      7  0.0    1.0
   END_GEOMETRY
   SETS
      [element and material definitions]
   END_SETS
END_GENERAL_DATA
```

```
GENERAL_DATA
    GEOMETRY   INTERPOLATE!  Parabolic 2D quadrilateral element with 9 nodes
        ELEM=1      SET=1     NODES=1  3  5  7  2  4  6  8  9
        NODE=1      X=0.0     Y=0.0
        NODE=3      X=1.0     Y=0.0
        NODE=5      X=1.0     Y=1.0
        NODE=7      X=0.0     Y=1.0
    END_GEOMETRY
    SETS
        [element and material definitions]
    END_SETS
END_GENERAL_DATA
```

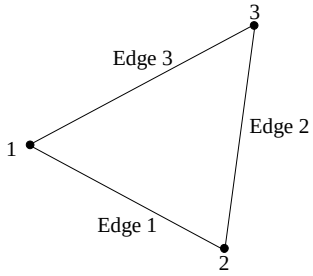
Available element:



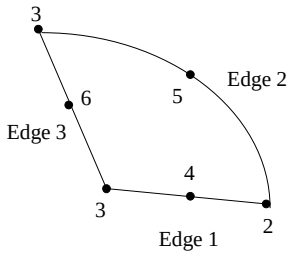
Linear1D element - Number of Nodes: 2



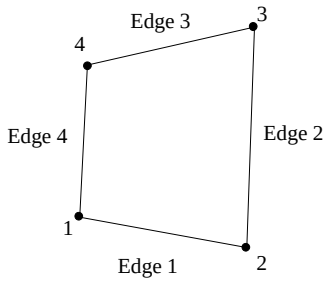
Linear1D element - Number of Nodes: 3



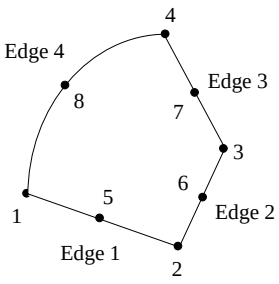
Linear triangular 2D element - Number of Nodes: 3



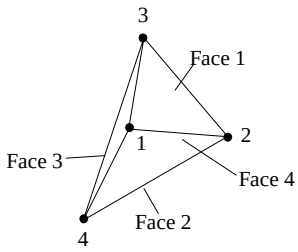
Parabolic triangular 2D element - Number of Nodes: 6



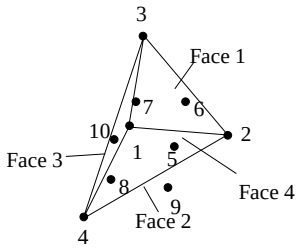
Quadrilateral 2D element - Number of Nodes: 4



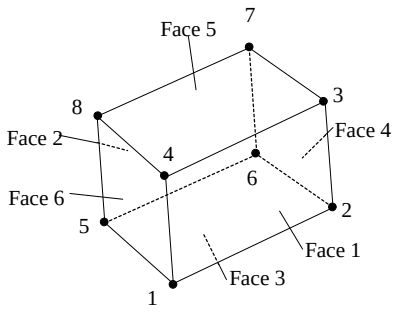
Parabolic quadrilateral 2D element - Number of Nodes: 8



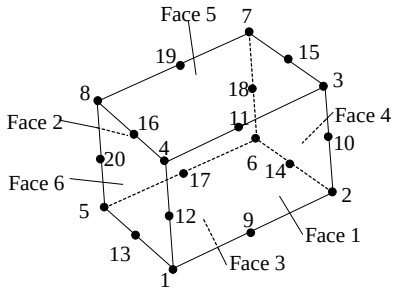
Solid Tetrahedral element - Number of Nodes: 4



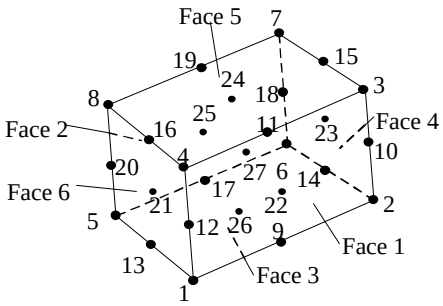
Solid Parabolic Tetrahedral element - Number of Nodes: 10



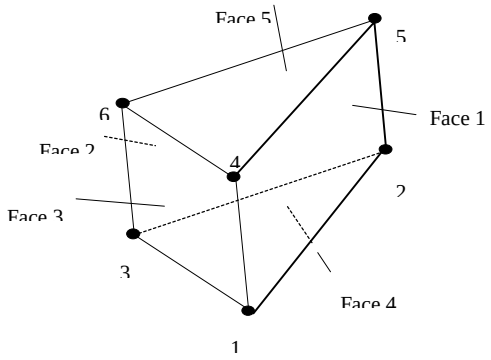
Solid Linear Brick element - Number of Nodes: 8



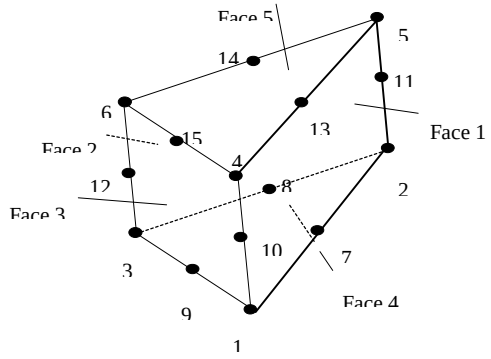
Solid Parabolic Brick element - Number of Nodes: 20



Solid Parabolic Brick element - Number of Nodes: 27



Solid Linear Prism element - Number of Nodes: 6



Solid Parabolic Prism element - Number of Nodes: 15

Figure 2. Nodes and edges of available elements

4.2 SETS

compulsory command

Description:

This command activates the input procedure for the element property data that is read immediately after this card.

Structure:

The sets data is made up of one set_data_block for each material set considered. This block contains all the element properties that are divided in the following 3 records:

(1) .SET main data that define the set

(2) ELEMENT_DATA definition of element properties

(3) MATERIAL_DATA definition of material properties

There should be NSETS of these set_data_blocks (defined in the DIMENSIONS command within the CONTROL_DATA block).

```
SETS
   SET=#    <parameters=#>
         [ELEMENT_DATA]     <parameters=#>
         [MATERIAL_DATA]    <parameters=#>
   END_SET_#
END_SETS
```

4.2.1 SET *<parameters=#>*

compulsory group of lines, one group for each NSETS set

Parameters:

SET=# Set identification number.

NODES=# Number of nodes.

TYPE=# Type of element identification number:

a) Generic SMALL STRAIN formulation elements

- 1 *mechanical* elements
- 2 *thermal* elements
- 3 *coupled thermo-mechanical* elements

b) Generic LARGE STRAIN formulation elements

- 4 *mechanical* large strain elements
- 5 *thermo-mechanical* large strain elements

c) Q1/P0 formulation specific for LINEAR QUADRILATERAL and HEXAHEDRAL element

- 6 *mechanical* SMALL strain elements
- 7 *mechanical* LARGE strain elements

d) MIXED U/P formulation specific for LINEAR TRIANGLES and TETRAHEDRA element

- 11 *mechanical* SMALL strain elements
- 13 *thermo-mechanical* SMALL strain elements
- 14 *mechanical* LARGE strain elements
- 15 *thermo-mechanical* LARGE strain elements

e) OPTIMIZED LINEAR TRIANGLES and TETRAHEDRA thermal element

- 12 *thermal* element

f) 2D/3D CONTACT FACE-TO-FACE JOINT element

- 21 *mechanical* contact elements
- 22 *thermal* contact elements
- 23 *thermo-mechanical* contact element

4.2.2 Element data *<parameters=#>*

The element properties necessary for a given element depend on the TYPE specified for it. See below for the specific data used by the available material element types.

TYPE=1 : Mechanical element: Small-strain Formulation

The following are the necessary properties to be input for the *mechanical* element within the framework of *small-strain* formulation. Any 2D/3D element type available in the COMET library can be used.

Parameters:

INT_RULE=# Integration rule to be used:

1	Quadrilateral or hexahedral elements: open rule
2	Quadrilateral or hexahedral elements: close rule
3	Triangular or tetrahedral elements: open rule
4	Triangular or tetrahedral elements: close rule
5	Irons rule
6	Prism elements: open rule
7	Prism elements: close rule

INT_POINTS=# Number of integration points per element, see tables below

MODEL=# Type of constitutive model:

11	J2-Visco-Plasticity
12	Linear Elasticity
15	Linear Visco-Elasticity
16	Isotropic Damage model
17	Tension-Compression Damage model
18	Concrete model (visco-elastic with damage)

THICKNESS=# Thickness. (Only for 2D)

DAMPM=# Rayleigh mass damping coefficient (for DYNAMIC problem),

DAMPK=# Rayleigh stiffness damping coefficient (for DYNAMIC problem).

TYPE=2 : Thermal element

The following are the necessary properties to be input for the *thermal* element. Any 2D/3D element type available in the COMET library can be used.

Parameters:

INT_RULE=# Integration rule to be used:

1	Quadrilateral or hexahedral elements: open rule
2	Quadrilateral or hexahedral elements: close rule
3	Triangular or tetrahedral elements: open rule
4	Triangular or tetrahedral elements: close rule
5	Irons rule
6	Prism elements: open rule
7	Prism elements: close rule

INT_POINTS=# Number of integration points per element, see tables below

MODEL=# Type of constitutive model:

21	Thermal model with or without phase change
25	Thermal model for the thermal partition of the different damage/concrete constitutive models with or without hydration heat generation

THICKNESS=# Thickness. (Only for 2D)

TYPE=3 : Thermo-Mechanical element: Small-strain Formulation

The following are the necessary properties to be input for the *thermo-mechanical* element within the framework of *small-strain* formulation. Any 2D/3D element type available in the COMET library can be used.

Parameters:

INT_RULE=# Integration rule to be used:

1	Quadrilateral or hexaedral elements: open rule
2	Quadrilateral or hexaedral elements: close rule
3	Triangular or tetrahedral elements: open rule
4	Triangular or tetrahedral elements: close rule
5	Irons rule
6	Prism elements: open rule
7	Prism elements: close rule

INT_POINTS=# Number of integration points per element, see tables below

MODEL=# Type of constitutive model:

31	Thermo-visco-elastic-J2-visco-plastic model with phase change
32	Thermo-linear-elastic model with phase change
35	Thermo-Linear Visco-Elasticity
36	Thermo Isotropic Damage model
37	Thermo Tension-Compression Damage model
38	Thermo-mechanical model for concrete aging/visco-elastic/damage

THICKNESS=# Thickness. (Only for 2D)

DAMPM=# Rayleigh mass damping coefficient (for DYNAMIC problem),

DAMPK=# Rayleigh stiffness damping coefficient (for DYNAMIC problem).

TYPE=4 : Mechanical element: Large-strain Formulation

The following are the necessary properties to be input for the *mechanical* element within the framework of *large-strain* formulation (total lagrangian updated configuration). Any 2D/3D element type available in the COMET library can be used.

Parameters:

INT_RULE=# Integration rule to be used:

1	Quadrilateral or hexahedral elements: open rule
2	Quadrilateral or hexahedral elements: close rule
3	Triangular or tetrahedral elements: open rule
4	Triangular or tetrahedral elements: close rule
5	Irons rule
6	Prism elements: open rule
7	Prism elements: close rule

INT_POINTS=# Number of integration points per element, see tables below

MODEL=# Type of constitutive model:

41	Large strain J2-visco-plasticity: principal stretches model
42	Large strain elasticity: principal stretches model
43	Large strain elasticity: Neo-Hooke model

THICKNESS=# Thickness. (Only for 2D)

DAMPM=# Rayleigh mass damping coefficient (for DYNAMIC problem),

DAMPK=# Rayleigh stiffness damping coefficient (for DYNAMIC problem).

TYPE=5 : Thermo-Mechanical element: Large-strain Formulation

The following are the necessary properties to be input for the *thermo-mechanical* element within the framework of *large-strain* formulation (total lagrangian updated configuration). Any 2D/3D element type available in the COMET library can be used.

Parameters:

INT_RULE=# Integration rule to be used:

1	Quadrilateral or hexahedral elements: open rule
2	Quadrilateral or hexahedral elements: close rule
3	Triangular or tetrahedral elements: open rule
4	Triangular or tetrahedral elements: close rule
5	Irons rule
6	Prism elements: open rule
7	Prism elements: close rule

INT_POINTS=# Number of integration points per element, see tables below

MODEL=# Type of constitutive model:

51	Thermo-J2-visco-plastic large strain model (principal stretches formulation)
52	Thermo-elastic large strain model (principal stretches formulation)

THICKNESS=# Thickness. (Only for 2D)

DAMPM=# Rayleigh mass damping coefficient (for DYNAMIC problem),

DAMPK=# Rayleigh stiffness damping coefficient (for DYNAMIC problem).

TYPE=6 : Q1/P0 Mechanical element: Small-strain Formulation

The following are the necessary properties to be input for the *mechanical* element within the framework of *small-strain* formulation. Q1/P0 algorithm is assumed to deal with incompressibility (a robust solution can be achieved even if a Poisson ratio very close to 0.5 is assumed). Only LINEAR QUADRILATERAL or HEXAHEDRAL elements can be used.

Parameters:

INT_RULE=# Integration rule to be used:

1 Quadrilateral or hexahedral elements: open rule
2 Quadrilateral or hexahedral elements: close rule

INT_POINTS=# Number of integration points per element, see tables below

MODEL=# Type of constitutive model:

11 J2-Visco-Plasticity
12 Linear Elasticity

THICKNESS=# Thickness. (Only for 2D)

DAMPM=# Rayleigh mass damping coefficient (for DYNAMIC problem),

DAMPK=# Rayleigh stiffness damping coefficient (for DYNAMIC problem).

TYPE=7 : Q1/P0 Mechanical element: Large-strain Formulation

The following are the necessary properties to be input for the *mechanical* element within the framework of *large-strain* formulation. Q1/P0 algorithm is assumed to deal with incompressibility (a robust solution can be achieved even if a Poisson ratio very close to 0.5 is assumed). Only LINEAR QUADRILATERAL or HEXAHEDRAL elements can be used.

Parameters:

INT_RULE=# Integration rule to be used:

1 Quadrilateral or hexahedral elements: open rule
2 Quadrilateral or hexahedral elements: close rule

INT_POINTS=# Number of integration points per element, see tables below

MODEL=# Type of constitutive model:

41 J2-Visco-Plasticity
42 Linear Elasticity
43 Large strain elasticity: Neo-Hooke model

THICKNESS=# Thickness. (Only for 2D)

DAMPM=# Rayleigh mass damping coefficient (for DYNAMIC problem),

DAMPK=# Rayleigh stiffness damping coefficient (for DYNAMIC problem).

TYPE=11 : Mixed U/P Mechanical element: Small-strain Formulation

The following are the necessary properties to be input for the *mechanical* element within the framework of *small-strain* formulation. Mixed U/P formulation is assumed to deal with incompressibility (even a value of 0.5 can be assigned to the Poisson ratio). Pressure is assumed as nodal variable together with the displacements field. The formulation is specific for LINEAR TRIANGLE or TETRAHERA elements. Only QUASI-STATIC analysis can be performed.

Parameters:

INT_RULE=# Integration rule to be used:

- 4 Triangular or tetrahedral elements: close rule

INT_POINTS=# Number of integration points per element, see tables below

MODEL=# Type of constitutive model:

- 11 J2-Visco-Plasticity
- 12 Linear Elasticity

THICKNESS=# Thickness. (Only for 2D)

TYPE=12 : Optimised Triangular / Tetrahedral Thermal element

The following are the necessary properties to be input for the *thermal* element. Memory database and computation have been optimised for LINEAR TRIANGLES or TETRAHEDRA elements.

Parameters:

INT_RULE=# Integration rule to be used:

- 4 Triangular or tetrahedral elements: close rule

INT_POINTS=# Number of integration points per element, see tables below

MODEL=# Type of constitutive model:

- 21 Thermal model with or without phase change
- 25,26,27,28 Thermal model for the thermal partition of the different damage/concrete constitutive models with or without hydration heat generation

THICKNESS=# Thickness. (Only for 2D)

TYPE=13 : Mixed U/P Thermo-Mechanical element: Small-strain Formulation

The following are the necessary properties to be input for the *thermo-mechanical* element within the framework of *small-strain* formulation. Mixed U/P formulation is assumed to deal with incompressibility (even a value of 0.5 can be assigned to the Poisson ratio). Pressure is assumed as nodal variable together with the displacements and temperature fields. The formulation is specific for LINEAR TRIANGLE or TETRAHERA elements. The mechanical partition can only deal with QUASI-STATIC analysis; thermal partition can be either steady-state or transient.

Parameters:

INT_RULE=# Integration rule to be used:

- 4 Triangular or tetrahedral elements: close rule

INT_POINTS=# Number of integration points per element, see tables below

MODEL=# Type of constitutive model:

- 31 Thermo-visco-elastic-J2-visco-plastic model with phase change
- 32 Thermo-linear-elastic model with phase change

THICKNESS=# Thickness. (Only for 2D)

TYPE=14 : Mixed U/P Mechanical element: Large-strain Formulation

The following are the necessary properties to be input for the *mechanical* element within the framework of *large-strain* formulation. Mixed U/P formulation is assumed to deal with incompressibility (even a value of 0.5 can be assigned to the Poisson ratio). Pressure is assumed as nodal variable together with the displacements field. The formulation is specific for LINEAR TRIANGLE or TETRAHERA elements. Only QUASI-STATIC analysis can be performed.

Parameters:

INT_RULE=# Integration rule to be used:

- 4 Triangular or tetrahedral elements: close rule

INT_POINTS=# Number of integration points per element, see tables below

MODEL=# Type of constitutive model:

- 41 J2-Visco-Plasticity
- 42 Linear Elasticity
- 43 Large strain elasticity: Neo-Hooke model

THICKNESS=# Thickness. (Only for 2D)

TYPE=21 : Mechanical Contact Face-to-Face Joint-element.

The following are the necessary properties to be input for the *mechanical contact face-to-face joint-element* within the framework of *small-displacements* formulation. The formulation is available for contact between 2D linear elements as LINEAR TRIANGLE or QUADRILATERAL elements and 3D LINEAR TETRAHEDRA. Penalty formulation is assumed.

Parameters:

MODEL=#	Type of constitutive model:
	81 penalty formulation
THICKNESS=#	Thickness. (Only for 2D)

TYPE=22 : Thermal Contact Face-to-Face Joint-element.

The following are the necessary properties to be input for the *thermal contact face-to-face joint-elemen*t. The formulation is available for contact between 2D linear elements as LINEAR TRIANGLE or QUADRILATERAL elements and 3D LINEAR TETRAHEDRA.

Parameters:

MODEL=#	Type of constitutive model:
	82 conduction/convection model
THICKNESS=#	Thickness. (Only for 2D)

TYPE=23 : Thermo-Mechanical Contact Face-to-Face Joint-element.

The following are the necessary properties to be input for the *thermo-mechanical contact face-to-face joint-element* within the framework of *small-displacements* formulation. The formulation is available for contact between 2D linear elements as LINEAR TRIANGLE or QUADRILATERAL elements and 3D LINEAR TETRAHEDRA. Penalty formulation is assumed for the mechanical partition.

Parameters:

MODEL=# Type of constitutive model:

83 mechanical partition: penalty formulation
thermal partition: conduction/convection model

THICKNESS=# Thickness. (Only for 2D)

Number of INTEGRATION POINTS

for different elements and integration rules:

	OPEN rule	**CLOSE rule**	**IRONS**
Linear TRIANG	1,**3**,4,6,7,13	**3**,4,6,7,10	-
Quadr TRIANG	1,3,4,6,**7**,13	3,4,6,**7**,10	-
Linear QUADR	1,**4**,9	1,**4**,9	-
Quadr QUADR	1,4,**9**	1,4,**9**	-
Linear TETRA	1,**4**,5,11,14	**4**,5,10,11,15,20	-
Quadr TETRA	1,4,5,**11**,14	4,5,10,**11**,15,20	-
Linear HEXA	1,**8**,27	1,**8**,27	**6**,14,15
Quadr HEXA	1,8,**27**	1,8,**27**	6,**14**,15
Linear PRISM	**6**	**6**	-
Quadr PRISM	**15**	**15**	-

4.2.3 Material data *<parameters=#>*

The material properties necessary for a given element depend on the MODEL specified for it. Material data may be constant or temperature dependent for some models. See below for the specific data used by the available material models.

Change of Phase structure:

- The change of phase properties must be input with the structure specified below.
- In each line must be input the properties of a particular change of phase: the first one is the *liquid-solid* transition.
- For the first change of phase (liquid-solid) it is possible to input the solid fraction as a function of temperature using a table in the temperature dependent properties structure (see temperature dependent properties section). If the solid fraction is not specified it will be considered a linear behaviour between liquidus and solidus temperature in the range [0:1].

```
PHASE_CHANGE
    CHANGE=1    <properties>, <qualifiers>
    CHANGE=2    <properties>, <qualifiers>
    …           …
    CHANGE=#    <properties>, <qualifiers>
END_PHASE_CHANGE
```

Temperature dependent properties:

- The input of temperature dependent properties is through user-defined tables.
- A temperature dependent property must not appear in the list of constant properties.
- A table composed of a series of pairs of values defines each property: the first value is the temperature and the second is the value of the property at that temperature.
- Each table must be input so to cover the full range of temperature used in the analysis.
- The values of the temperature defining the table for each property do not need to be the same. Piece-wise linear or cubic-splines interpolation can be used for each table independently.
- The function defined by the pairs must be continuous and possibly smoothed. Discontinuous functions must be regularised.

Structure of the temperature-dependent-data:

The temperature_dependent_data corresponding to the set is an extra dataset to be included in the set_data after the material_data as follows:

```
TDEPENDENT_DATA
    PROP_1:    <qualifier>
      temp_1       value_1
      temp_2       value_2
    END_PROP_1
    PROP_2:    <qualifier>
      temp_3       value_3
      temp_4       value_4
    END_PROP_2
END_TDEPENDENT_DATA
```

Qualifier for the temperature-dependent-data:

PIECE_WISE	The values given in the table that defines the considered property in the range of temperature of the analysis are interpolated with a piece_wise function.
SPLINE	The values given in the table that defines the considered property in the range of temperature of the analysis are interpolated with a cubic-spline function.

Default for the temperature-dependent-data:

PIECE_WISE

Examples:

```
TDEPENDENT_DATA
        CONDU:  PIECE_WISE
            273.0  1.5
            400.0  2.1
        END_CONDU
        YOUNG:  SPLINE
            273.0  20000.
            350.0  22000.
            400.0  25000.
        END_YOUNG
END_TDEPENDENT_DATA
```

MODEL=11 : J2-Elasto-Visco-Plasticity

The following are the necessary properties to be input to fully define the *J2 Elasto-Plastic Model*. The model may include isotropic and kinematic hardening.

Parameters:

DENSI=#	Density (compulsory parameter only for dynamic analysis)
YOUNG=#	Elastic modulus.
POISS=#	Poisson's ratio.
YEINI=#	Initial flow stress
YEFIN=#	Saturation flow stress
YEPOW=#	Saturation hardening law exponent
LINHR=#	Linear hardening coefficient
KHARD=#	Linear kinematic hardening
NLKHD=#	Non-linear kinematic hardening
VISCO=#	Viscosity for visco-plastic model
EXPVIS=#	Exponent for plastic viscosity

The isotropic hardening function is defined with the following laws:

1. Perfect plasticity: the elastic limit is YEINI (YEFIN= YEINI)

   $$\sigma_Y(\zeta) = YEINI$$

2. Linear isotropic hardening:

   $$\sigma_Y(\zeta) = YEINI + LINHR \cdot \zeta$$

3. Saturation law:

   $$\sigma_Y(\zeta) = YEINI + [YEFIN - YEINI] \cdot [1 - \exp(-YEPOW \cdot \zeta)]$$

4. Sturation law + linear hardening:

   $$\sigma_Y(\zeta) = YEINI + [YEFIN - YEINI] \cdot [1 - \exp(-YEPOW \cdot \zeta)] + LINHR \cdot \zeta$$

where ζ is the equivalent plastic strain.

Example:

```
SETS
   SET=#     <parameters=#>
         [ELEMENT_DATA]      <parameters=#>
         MATERIAL_DATA:    YOUNG=20000.   POISS=0.23   /
                           YEINI=260.   YEFIN=270.    YEPOW=16  LINHR=100.
   END_SET_#
END_SETS
```

MODEL=12 : Elasticity

The following are the necessary properties to be input to fully define the *Elastic Model.*

Parameters:

DENSI=#	Density (compulsory parameter only for dynamic analysis)
YOUNG=#	Elastic modulus.
POISS=#	Poisson's ratio.

Example:

```
SETS
   SET=#    <parameters=#>
         [ELEMENT_DATA]      <parameters=#>
         MATERIAL_DATA:    YOUNG=20000.   POISS=0.3
   END_SET_#
END_SETS
```

MODEL=15 : Linear Visco-elasticity model

The following are the necessary properties to be input to fully define the *linear visco-elasticity model.*

Elastic parameters:

DENSI=#	Density
YOUNG=#	Elastic modulus
POISS=#	Poisson's ratio

Visco-elastic parameters:

NMAXW=#	Number of Maxwell chains (NAMXW=0 -- 5)
	(optional, default: NMAXW=0)
ELAS(i)=#	Elasticity modulus ratio for chain (i)
RETA(i)=#	Retardation time for chain (i)

Example 1: Elastic

```
SETS
   SET=#    <parameters=#>
         [ELEMENT_DATA]     <parameters=#>
         MATERIAL_DATA:   DENSI=2000      YOUNG=20E+09.   POISS=0.
   END_SET_#
END_SETS
```

Example 2: Visco-elastic (2 Maxwell elements)

```
SETS
   SET=#    <parameters=#>
         [ELEMENT_DATA]     <parameters=#>
         MATERIAL_DATA:   DENSI=2000      YOUNG=20E+09.   POISS=0.2          /
                          NMAXW=2                                            /
                          ELAS1=0.5      RETA1=10   ELAS2=0.2      RETA2=100
   END_SET_#
END_SETS
```

MODEL=16 : Isotropic Damage model

The following are the necessary properties to be input to fully define the *isotropic damage model.*

Elastic parameters:

DENSI=#	Density
YOUNG=#	Elastic modulus
POISS=#	Poisson's ratio

Damage parameters:

STREN=#	Uniaxial strength
ILAWT=#	Type of hardening/softening law:
	ILAWT=0 → linear hardening/softening law
	ILAWT=1 → exponential hardening/softening law
HFACT=#	Linear hardening parameter (only for ILAWT=0)
GFRAC=#	Fracture energy (to define a regularized softening behaviour)
RETAT=#	Retardation time for rate-dependent damage
	(optional, default: RETAT=0.0)
EXPDT=#	Exponent for rate-dependent damage
	(optional, default: EXPDT=1.0)

Visco-elastic parameters:

NMAXW=#	Number of Maxwell chains (NAMXW=0 -- 5)
	(optional, default: NMAXW=0)
ELAS(i)=#	Elasticity modulus ratio for chain (i)
RETA(i)=#	Retardation time for chain (i)

Example-1: Damage with linear hardening (rate independent)

```
SETS
   SET=#    <parameters=#>
         [ELEMENT_DATA]     <parameters=#>
         MATERIAL_DATA:   DENSI=2000      YOUNG=20E+09.   POISS=0.2  /
                          STREN=1E+06     ILAWT=0         HFACT=0.1
   END_SET_#
END_SETS
```

Example-2: Damage with exponential softening (rate independent)

```
SETS
   SET=#    <parameters=#>
         [ELEMENT_DATA]     <parameters=#>
         MATERIAL_DATA:   DENSI=2000      YOUNG=20E+09.   POISS=0.2  /
                          STREN=1E+06     ILAWT=1         GFRAC=2.0E+02
   END_SET_#
END_SETS
```

Example-3: Rate dependent damage

```
SETS
   SET=#    <parameters=#>
         [ELEMENT_DATA]     <parameters=#>
         MATERIAL_DATA:   DENSI=2000      YOUNG=20E+09.   POISS=0.2        /
                          STREN=1E+06     ILAWT=1         GFRAC=2.0E+02    /
                          RETAT= 100
   END_SET_#
END_SETS
```

Example-4: Visco-elastic (1 Maxwell element) / Rate dependent damage

```
SETS
   SET=#    <parameters=#>
         [ELEMENT_DATA]     <parameters=#>
         MATERIAL_DATA:   DENSI=2000      YOUNG=20E+09.   POISS=0.2        /
                          STREN=1E+06     ILAWT=1         GFRAC=2.0E+02    /
                          RETAT= 100                                       /
                          NMAXW=1         ELAS1=0.5       RETA1=10
   END_SET_#
END_SETS
```

MODEL=17 : Tension-Compression Damage model

The following are the necessary properties to be input to fully define the *tension-compression damage model.*

Elastic parameters:

DENSI=#	Density
YOUNG=#	Elastic modulus
POISS=#	Poisson's ratio

Tensile damage parameters:

STREN=#	Tensile strength
ILAWT=#	Type of hardening/softening law:
	ILAWT=0 → linear hardening/softening law
	ILAWT=1 → exponential hardening/softening law
HFACT=#	Linear hardening parameter (only for ILAWT=0)
GFRAC=#	Tensile fracture energy (to define a regularized softening behaviour)
RETAT=#	Retardation time for rate-dependent tensile damage
	(optional, default: RETAT=0.0)
EXPDT=#	Exponent for rate-dependent tensile damage
	(optional, default: EXPDT=1.0)

Compressive damage parameters:

RACTS=#	Compressive/Tensile strengths ratio (compulsory)
CDAMG=#	Compressive damage threshold/Compressive strength ratio
	(optional, default: CDAMG=1.0)
GFCOM=#	Compressive fracture energy (to define a regularized softening behaviour)
	(optional, default: GFCOM=GRFAC*$(RACTS)^2$)
EPSUC=#	Peak compressive strain
	(optional, default= (2.0- CDAMG)* RACTS*STREN/YOUNG)
RETAC=#	Retardation time for rate-dependent compressive damage
	(optional, default: RETAC=RETAT)
EXPDC=#	Exponent for rate-dependent compressive damage
	(optional, default: EXPDC=1.0)

Visco-elastic parameters:

NMAXW=#	Number of Maxwell chains (NAMXW=0 -- 5)
	(optional, default: NMAXW=0)
ELAS(i)=#	Elasticity modulus ratio for chain (i)
RETA(i)=#	Retardation time for chain (i)

Example-1: Damage in tension and compression. Linear hardening

```
SETS
   SET=#    <parameters=#>
         [ELEMENT_DATA]     <parameters=#>
         MATERIAL_DATA:  DENSI=2000     YOUNG=20E+09.  POISS=0.2  /
                         STREN=1E+06    ILAWT=0        HFACT=0.1 /
                         RACTS=1
   END_SET_#
END_SETS
```

Example-2: Damage in tension and compression (rate independent)

```
SETS
   SET=#    <parameters=#>
         [ELEMENT_DATA]     <parameters=#>
         MATERIAL_DATA:  DENSI=2000     YOUNG=20E+09.  POISS=0.2  /
                         STREN=1E+06    ILAWT=1        GFRAC=2.0E+02  /
                         RACTS=10       CDAMG =0.3     GFCOM=2.0E+04
   END_SET_#
END_SETS
```

Example-3: Damage in tension and compression (rate dependent)

```
SETS
   SET=#    <parameters=#>
         [ELEMENT_DATA]     <parameters=#>
         MATERIAL_DATA:  DENSI=2000     YOUNG=20E+09.  POISS=0.2          /
                         STREN=1E+06    ILAWT=1        GFRAC=2.0E+02  /
                         RETAT= 100                                   /
                         RACTS=10       CDAMG =0.3     GFCOM=2.0E+04  /
                         RETAC= 200
   END_SET_#
END_SETS
```

Example-4: Visco-elastic (2 Maxwell elements) /
Damage in tension and compression (rate dependent)

```
SETS
   SET=#     <parameters=#>
         [ELEMENT_DATA]      <parameters=#>
         MATERIAL_DATA:   DENSI=2000      YOUNG=20E+09.   POISS=0.2          /
                          STREN=1E+06     ILAWT=1         GFRAC=2.0E+02      /
                          RETAT= 100                                         /
                          RACTS=10        CDAMG =0.3      GFCOM=2.0E+04      /
                          RETAC= 200                                         /
                          NMAXW=2                                            /
                          ELAS1=0.3       RETA1=10   ELAS2=0.2    RETA2=100
   END_SET_#
END_SETS
```

MODEL=18 : Concrete model : visco-elastic with damage

The following are the necessary properties to be input to fully define the *concrete model (visco-elastic with damage).*

Elastic parameters:

DENSI=#	Density
YOUNG=#	Elastic modulus
POISS=#	Poisson's ratio

Compressive damage parameters:

STREC=#	Compressive strength (final value)
EPSUC=#	Peak compressive strain (optional)
YEIDC=#	Compressive damage threshold stress
GFRAC=#	Compressive fracture energy

Tensile damage parameters:

STRET=#	Tensile strength (final value)
GFRAT=#	Tensile fracture energy

Visco-elastic parameters:

NMAXW=#	Number of Maxwell chains (0-5) (optional)
ELAS(i)=#	Elasticity modulus ratio for chain (i)
RETA(i)=#	Retardation time for chain (i)

Micro pre-stress parameters:

CFLOW=#	Constant for evolution of micro-prestress (optional)
RFLOW=#	Relaxation time for evolution of micro-prestress (optional)

Example-1: Damage in tension and compression

```
SETS
   SET=#    <parameters=#>
         [ELEMENT_DATA]     <parameters=#>
         MATERIAL_DATA:   DENSI=2000  YOUNG=20000.   POISS=0.3   /
                          STREC=300     YEIDC=30       GFRAC=1000   /
                          STRET=25      GFRAT=100
   END_SET_#
END_SETS
```

Example-2: Visco-elastic (2 Maxwell elements + micro-prestress) / Damage in tension and compression

```
SETS
   SET=#    <parameters=#>
         [ELEMENT_DATA]     <parameters=#>
         MATERIAL_DATA:   DENSI=2000  YOUNG=20000.   POISS=0.3   /
                          STREC=300     YEIDC=30       GFCOM=1000   /
                          STRET=25      GFRAC=100   /
                          NMAXW=2    /
                          ELAS1=0.5     ELAS2=0.3      RETA1=100     RETA2=10.   /
                          CFLOW=1.0    RFLOW=2E6
   END_SET_#
END_SETS
```

MODEL=21 : Thermal Model with or without change of phase

The following are the necessary properties to be input to fully define the *Thermal Model with multiphase changes*. Most of the properties can be considered as temperature dependent (see below). The model may also reproduce solidification processes with multiphase changes (see below).

Parameters:

DENSI=#	Density (at reference temperature)
SHEAT=#	Specific heat
CONDU=#	Conductivity

Change of phase properties:

LATEN=#	Latent heat released.
THOTP=#	Hot phase temperature.
TCOLD=#	Cold phase temperature.

Temperature dependent properties:

The following properties can be defined as temperature dependent following the procedure explained above. Note that the LATENT_HEAT can only be input as temperature dependent. The Latent _heat curve can be constructed to represent more than one phase change (liquid-solid, solid-solid, etc..).

SHEAT	Specific heat
CONDU	Conductivity
FRSOL	Solid-fraction (only for phase change analysis)

Example:

```
SETS
   SET=#    <parameters=#>
         [ELEMENT_DATA]      <parameters=#>
         MATERIAL_DATA:   DENSITY=7800.   SHEAT=100.0
         PHASE_CHANGE
               CHANGE=1      LATEN=3.0E5   THOTP=660.0   TCOLD=559.0
         END_PHASE_CHANGE
   END_SET_#
END_SETS
```

MODEL=25 : Thermal Model with or without hydration heat generation

The following are the necessary properties to be input to fully define the *Thermal Model with hydration heat generation.*

Parameters:

DENSI=#	Density
SHEAT=#	Specific heat
CONDU=#	Conductivity

Hydration properties:

RINFI=#	Final hydration degree
LATEN=#	Hydration latent heat
CONSK=#	k/η constant
EXPON=#	Exponent for η
ENERG=#	Activation energy / ideal-gas constant
AFFI0=#	Initial affinity / k

Example:

```
SETS
   SET=#    <parameters=#>
         [ELEMENT_DATA]     <parameters=#>
         MATERIAL_DATA:   DENSITY=2200.   SHEAT=1000.0   CONDU=6000 /
                          RINFI=0.7    LATEN=6E7    CONSK=4E6  /
                          EXPON=6    ENERG=4000   AFFI0=1E-10
   END_SET_#
END_SETS
```

MODEL=31 : Thermal-Visco-elastic- Visco-Plastic with or without phase change

The following are the necessary properties to be input to fully define the *Thermo-visco-elastic-J2 visco-plastic model with multiphase changes*. The model may include both isotropic and kinematic hardening and thermal softening. Most of the properties can be considered as temperature dependent (see below). The model may also reproduce solidification processes with multiphase changes (see below).

Parameters:

DENSI=#	Density (at reference temperature)
YOUNG=#	Elastic modulus
POISS=#	Poisson's ratio
ALPHA=#	Thermal expansion coefficient
SHEAT=#	Specific heat
CONDU=#	Conductivity
YEINI=#	Initial flow stress
YEFIN=#	Saturation flow stress
YEPOW=#	Saturation hardening law exponent
LINHR=#	Linear hardening coefficient
KHARD=#	Linear kinematic hardening
NLKHD=#	Non-linear kinematic hardening
VISCO=#	Viscosity for Norton viscous model (liquidus state) and viscosity for visco-plastic model (solidus state).
EXPVIS=#	Exponent for plastic viscosity

The isotropic hardening function is defined with the following saturation law (see model 11):

$$\sigma_Y(\zeta,\Theta) = YEINI(\Theta) + [YIEFIN(\Theta) - YEINI(\Theta)]\cdot[1 - \exp(-YEPOW\cdot\zeta)] + LINHR(\Theta)\cdot\zeta$$

where Θ is the current absolute temperature and ζ is the equivalent plastic strain.

Change of phase properties:

LATEN=#	Latent heat released.
THOTP=#	Hot phase temperature.
TCOLD=#	Cold phase temperature.
PSTRA=#	Straining due to change of phase (shrinkage)

Temperature dependent properties:

The following properties can be defined as temperature dependent following the procedure explained above.

YOUNG	Elastic modulus
ALPHA	Thermal expansion coefficient
SHEAT	Specific heat
CONDU	Conductivity
YEINI	Initial flow stress
YEFIN	Saturation flow stress
YEPOW	Saturation hardening law exponent
LINHR	Linear hardening coefficient
KHARD	Linear kinematic hardening
NLKHD	Non-linear kinematic hardening
VISCO	Viscosity table
EXPVI	Exponent for plastic viscosity
FRSOL	Solid-fraction (only for phase change analysis)

Example:

```
SETS
   SET=#     <parameters=#>
      [ELEMENT_DATA]          <parameters=#>
      MATERIAL_DATA:        DENSITY=7800.   POISS=0.23    SHEAT=100.0  /
                            ALPHA=1.E-5     VISCO=100.0E6
                            YEINI=260.  YEFIN=270.   YEPOW=16  LINHR=100.  /
                            KHARD=150
      PHASE_CHANGE
         CHANGE=1   LATEN=3.0E5   THOTP=660.0   TCOLD=550.0   PSTRA=0.02
      END_PHASE_CHANGE
      TDEPENDENT_DATA
         CONDU, PIECE_WISE
            20.0   1.5
            700.0  2.1
         END_CONDU
         YOUNG, SPLINE
            20.0   20000
            350.0  22000
            700.0  25000
         END_YOUNG
      END_TDEPENDENT_DATA
   END_SET_#
END_SETS
```

MODEL=32 : Thermo-elastic with or without phase change

The following are the necessary properties to be input to fully define the *Thermo-elastic model with multiphase changes*. Most of the properties can be considered as temperature dependent (see below). The model may also reproduce solidification processes with multiphase changes (see below).

Parameters:

DENSI=#	Density (at reference temperature)
YOUNG=#	Elastic modulus
POISS=#	Poisson's ratio
ALPHA=#	Thermal expansion coefficient
SHEAT=#	Specific heat
CONDU=#	Conductivity
VISCO=#	Viscosity for Norton viscous model (liquidus state).

Change of phase properties:

LATEN=#	Latent heat released.
THOTP=#	Hot phase temperature.
TCOLD=#	Cold phase temperature.
PSTRA=#	Straining due to change of phase (shrinkage)

Temperature dependent properties:

The following properties can be defined as temperature dependent following the procedure explained above.

YOUNG	Elastic modulus
ALPHA	Thermal expansion coefficient
SHEAT	Specific heat
CONDU	Conductivity
VISCO	Viscosity table
FRSOL	Solid-fraction (only for phase change analysis)

Example:

```
SETS
   SET=#     <parameters=#>
      [ELEMENT_DATA]          <parameters=#>
      MATERIAL_DATA:        DENSITY=7800.   POISS=0.23    SHEAT=100.0  /
                            ALPHA=1.E-5     VISCO=100.0E6
      PHASE_CHANGE
         CHANGE=1    LATEN=3.0E5    THOTP=660.0    TCOLD=550.0   PSTRA=0.02
      END_PHASE_CHANGE
      TDEPENDENT_DATA
         CONDU, PIECE_WISE
            20.0    1.5
            700.0  2.1
         END_CONDU
         YOUNG, SPLINE
            20.0    20000
            350.0  22000
            700.0  25000
         END_YOUNG
      END_TDEPENDENT_DATA
   END_SET_#
END_SETS
```

MODEL=35 : Thermo-Linear Visco-elasticity model

The following are the necessary properties to be input to fully define the *thermo- linear visco-elasticity model.*

Thermal parameters:

SHEAT=#	Specific heat
CONDU=#	Conductivity
ALPHA=#	Thermal expansion coefficient
TEREF=#	Reference temperature

Elastic parameters:

DENSI=#	Density
YOUNG=#	Elastic modulus
POISS=#	Poisson's ratio

Visco-elastic parameters:

NMAXW=#	Number of Maxwell chains (NAMXW=0 -- 5)
	(optional, default: NMAXW=0)
ELAS(i)=#	Elasticity modulus ratio for chain (i)
RETA(i)=#	Retardation time for chain (i)

Example 1: Elastic

```
SETS
   SET=#    <parameters=#>
         [ELEMENT_DATA]     <parameters=#>
         MATERIAL_DATA:   DENSI=2000      YOUNG=20E+09.   POISS=0.  /
                          SHEAT=100.0  CONDU= 100.  ALPHA=1.E-5   TEREF=20.

   END_SET_#
END_SETS
```

Example 2: Visco-elastic (2 Maxwell elements)

```
SETS
   SET=#    <parameters=#>
         [ELEMENT_DATA]     <parameters=#>
         MATERIAL_DATA:   DENSI=2000      YOUNG=20E+09.   POISS=0.2      /
                          SHEAT=100.0  CONDU= 100.  ALPHA=1.E-5   TEREF=20  /
                          NMAXW=2                                          /
                          ELAS1=0.5      RETA1=10   ELAS2=0.2      RETA2=100
   END_SET_#
END_SETS
```

MODEL=36 : Thermo- Isotropic Damage model

The following are the necessary properties to be input to fully define the *thermo-isotropic damage model.*

Thermal parameters:

SHEAT=#	Specific heat
CONDU=#	Conductivity
ALPHA=#	Thermal expansion coefficient
TEREF=#	Reference temperature

Elastic parameters:

DENSI=#	Density
YOUNG=#	Elastic modulus
POISS=#	Poisson's ratio

Damage parameters:

STREN=#	Tensile strength
ILAWT=#	Type of hardening/softening law:
	ILAWT=0 → linear hardening/softening law
	ILAWT=1 → exponential hardening/softening law
HFACT=#	Linear hardening parameter (only for ILAWT=0)
GFRAC=#	Tensile fracture energy (to define a regularized softening behaviour)
RETAT=#	Retardation time for rate-dependent tensile damage
	(optional, default: RETAT=0.0)

Visco-elastic parameters:

NMAXW=#	Number of Maxwell chains (NAMXW=0 -- 5)
	(optional, default: NMAXW=0)
ELAS(i)=#	Elasticity modulus ratio for chain (i)
RETA(i)=#	Retardation time for chain (i)

Example-1: Damage with linear hardening (rate independent)

```
SETS
   SET=#    <parameters=#>
         [ELEMENT_DATA]     <parameters=#>
         MATERIAL_DATA:   DENSI=2000      YOUNG=20E+09.   POISS=0.2  /
                          SHEAT=100.0  CONDU= 100.  ALPHA=1.E-5   TEREF=20  /
                          STREN=1E+06    ILAWT=0           HFACT=0.1
   END_SET_#
END_SETS
```

Example-2: Damage with exponential softening (rate independent)

```
SETS
   SET=#    <parameters=#>
         [ELEMENT_DATA]     <parameters=#>
         MATERIAL_DATA:   DENSI=2000      YOUNG=20E+09.   POISS=0.2  /
                          SHEAT=100.0  CONDU= 100.  ALPHA=1.E-5   TEREF=20  /
                          STREN=1E+06    ILAWT=1           GFRAC=2.0E+02
   END_SET_#
END_SETS
```

Example-3: Rate dependent damage

```
SETS
   SET=#    <parameters=#>
         [ELEMENT_DATA]     <parameters=#>
         MATERIAL_DATA:   DENSI=2000      YOUNG=20E+09.   POISS=0.2          /
                          SHEAT=100.0  CONDU= 100.  ALPHA=1.E-5   TEREF=20  /
                          STREN=1E+06    RETAT= 100
   END_SET_#
END_SETS
```

Example-4: Visco-elastic (1 Maxwell element) / Rate dependent damage

```
SETS
   SET=#    <parameters=#>
         [ELEMENT_DATA]     <parameters=#>
         MATERIAL_DATA:   DENSI=2000      YOUNG=20E+09.   POISS=0.2          /
                          SHEAT=100.0  CONDU= 100.  ALPHA=1.E-5   TEREF=20   /
                          STREN=1E+06    ILAWT=1           GFRAC=2.0E+02   /
                          RETAT= 100                                       /
                          NMAXW=1        ELAS1=0.5       RETA1=10
   END_SET_#
END_SETS
```

MODEL=37 : Thermo-Tension-Compression Damage model

The following are the necessary properties to be input to fully define the *thermo-tension-compression damage model.*

Thermal parameters:

SHEAT=#	Specific heat
CONDU=#	Conductivity
ALPHA=#	Thermal expansion coefficient
TEREF=#	Reference temperature

Elastic parameters:

DENSI=#	Density
YOUNG=#	Elastic modulus
POISS=#	Poisson's ratio

Tensile damage parameters:

STREN=#	Tensile strength
ILAWT=#	Type of hardening/softening law:
	ILAWT=0 → linear hardening/softening law
	ILAWT=1 → exponential hardening/softening law
HFACT=#	Linear hardening parameter (only for ILAWT=0)
GFRAC=#	Tensile fracture energy (to define a regularized softening behaviour)
RETAT=#	Retardation time for rate-dependent tensile damage
	(optional, default: RETAT=0.0)

Compressive damage parameters:

RACTS=#	Compressive/Tensile strengths ratio (compulsory)
CDAMG=#	Compressive damage threshold/Compressive strength ratio
	(optional, default: CDAMG=1.0)
GFCOM=#	Compressive fracture energy (to define a regularized softening behaviour)
	(optional, default: GFCOM=GRFAC*$(RACTS)^2$)
EPSUC=#	Peak compressive strain
	(optional, default= (2.0- CDAMG)* RACTS*STREN/YOUNG)
RETAC=#	Retardation time for rate-dependent compressive damage
	(optional, default: RETAC=RETAT)

Visco-elastic parameters:

NMAXW=#	Number of Maxwell chains (NAMXW=0 -- 5)
	(optional, default: NMAXW=0)
ELAS(i)=#	Elasticity modulus ratio for chain (i)
RETA(i)=#	Retardation time for chain (i)

Example-1: Damage in tension and compression. Linear hardening

```
SETS
   SET=#    <parameters=#>
         [ELEMENT_DATA]     <parameters=#>
         MATERIAL_DATA:   DENSI=2000      YOUNG=20E+09.   POISS=0.2  /
                          SHEAT=100.0  CONDU= 100.  ALPHA=1.E-5   TEREF=20  /
                          STREN=1E+06   ILAWT=0           HFACT=0.1 /
                          RACTS=1
   END_SET_#
END_SETS
```

Example-2: Damage in tension and compression (rate independent)

```
SETS
   SET=#    <parameters=#>
         [ELEMENT_DATA]     <parameters=#>
         MATERIAL_DATA:   DENSI=2000      YOUNG=20E+09.   POISS=0.2  /
                          SHEAT=100.0  CONDU= 100.  ALPHA=1.E-5   TEREF=20  /
                          STREN=1E+06   ILAWT=1           GFRAC=2.0E+02  /
                          RACTS=10      CDAMG =0.3        GFCOM=2.0E+04
   END_SET_#
END_SETS
```

Example-3: Damage in tension and compression (rate dependent)

```
SETS
   SET=#    <parameters=#>
         [ELEMENT_DATA]     <parameters=#>
         MATERIAL_DATA:   DENSI=2000      YOUNG=20E+09.   POISS=0.2          /
                          SHEAT=100.0  CONDU= 100.  ALPHA=1.E-5   TEREF=20  /
                          STREN=1E+06   ILAWT=1           GFRAC=2.0E+02  /
                          RETAT= 100                                     /
                          RACTS=10      CDAMG =0.3        GFCOM=2.0E+04  /
                          RETAC= 200
   END_SET_#
END_SETS
```

Example-4: Visco-elastic (2 Maxwell elements) /
Damage in tension and compression (rate dependent)

```
SETS
   SET=#     <parameters=#>
         [ELEMENT_DATA]      <parameters=#>
         MATERIAL_DATA:    DENSI=2000       YOUNG=20E+09.   POISS=0.2            /
                           SHEAT=100.0  CONDU= 100.  ALPHA=1.E-5   TEREF=20  /
                           STREN=1E+06    ILAWT=1            GFRAC=2.0E+02   /
                           RETAT= 100                                        /
                           RACTS=10        CDAMG =0.3        GFCOM=2.0E+04   /
                           RETAC= 200                                        /
                           NMAXW=2                                           /
                           ELAS1=0.3       RETA1=10   ELAS2=0.2    RETA2=100
   END_SET_#
END_SETS
```

MODEL=38 : Thermo-mechanical model for concrete: aging/visco-elastic/damage

The following are the necessary properties to be input to fully define the *thermo-mechanical model for concrete aging/visco-elastic/damage*

Elastic parameters:

DENSI=#	Density
YOUNG=#	Elastic modulus
POISS=#	Poisson's ratio

Compressive damage parameters:

STREC=#	Compressive strength (final value)
EPSUC=#	Peak compressive strain (optional)
YEIDC=#	Compressive damage threshold stress
GFRAC=#	Compressive fracture energy

Tensile damage parameters:

STRET=#	Tensile strength (final value)
GFRAT=#	Tensile fracture energy

Visco-elastic parameters (optional):

NMAXW=#	Number of Maxwell chains (0-5)
ELAS(i)=#	Elasticity modulus ratio for chain (i)
RETA(i)=#	Retardation time for chain (i)

Micro pre-stress parameters (optional):

CFLOW=#	Constant for evolution of micro-prestress (optional)
RFLOW=#	Relaxation time for evolution of micro-prestress (optional)

Thermal properties:

SHEAT=#	Specific heat
CONDU=#	Conductivity

Coupling properties:

ALPHA=#	Thermal expansion coefficient
BETHA=#	Hydration shrinkage coefficient
TEREF=#	Reference temperature (optional)

Hydration properties (optional):

RINFI=#	Final hydration degree
LATEN=#	Hydration latent heat
CONSK=#	k/η constant
EXPON=#	Exponent for η
ENERG=#	Activation energy / ideal-gas constant
AFFI0=#	Initial affinity / k

Aging properties (optional):

RCRIT=#	Critical hydration degree
FCRIT=#	Critical compressive strength
RINTE=#	Intermediate hydration degree
FINTE=#	Intermediate compressive strength
TEMAN=#	Exponent for thermal aging effect
TEMAT=#	Reference temperature for thermal aging effect
CUBIC	Flag for cubic interpolation of aging function

Example-1: Thermo-mechanical with damage in tension and compression

```
SETS
   SET=#    <parameters=#>
         [ELEMENT_DATA]     <parameters=#>
         MATERIAL_DATA:   DENSI=2000  YOUNG=20000.   POISS=0.3   /
                          STREC=300     YEIDC=30        GFRAC=1000   /
                          STRET=25     GFRAT=100  /
                          SHEAT=1000.0   CONDU=6000 /
                          ALPHA=1E-6     TEREF=20.0
   END_SET_#
END_SETS
```

Example-2: Thermo visco-elastic (2 Maxwell elements + micro-prestress) / Damage in tension and compression

```
SETS
   SET=#    <parameters=#>
         [ELEMENT_DATA]     <parameters=#>
         MATERIAL_DATA:   DENSI=2000  YOUNG=20000.   POISS=0.3   /
                          STREC=300     YEIDC=30        GFRAC=1000   /
                          STRET=25     GFRAT=100  /
                          NMAXW=2    /
                          ELAS1=0.5     ELAS2=0.3      RETA1=100     RETA2=10.   /
                          CFLOW=1.0    RFLOW=2E6
                          SHEAT=1000.0   CONDU=6000 /
                          ALPHA=1E-6     TEREF=20.0
   END_SET_#
END_SETS
```

Example-3: Thermo-damage in tension and compression / Aging

```
SETS
   SET=#    <parameters=#>
         [ELEMENT_DATA]     <parameters=#>
         MATERIAL_DATA:   DENSI=2000  YOUNG=20000.   POISS=0.3   /
                          STREC=300     YEIDC=30        GFRAC=1000   /
                          STRET=25     GFRAT=100  /
                          SHEAT=1000.0   CONDU=6000 /
                          ALPHA=1E-6     TEREF=20.0  /
                          RINFI=0.7    LATEN=6E7    CONSK=4E6  /
                          EXPON=6    ENERG=4000   AFFI0=1E-10  /
                          RCRIT=0.2   FCRIT=15.0   RINTE=0.5   FINTE=100   CUBIC  /
                          TEMAT=20.0   TEMAN=0.5   BETHA=1E-6
   END_SET_#
END_SETS
```

Example-4: Thermo visco-elastic (2 Maxwell elements + micro-prestress) /
Damage in tension and compression /
Aging

```
SETS
   SET=#    <parameters=#>
         [ELEMENT_DATA]     <parameters=#>
         MATERIAL_DATA:   DENSI=2000  YOUNG=20000.   POISS=0.3   /
                          STREC=300     YEIDC=30        GFRAC=1000   /
                          STRET=25     GFRAT=100  /
                          NMAXW=2    /
                          ELAS1=0.5     ELAS2=0.3      RETA1=100     RETA2=10.   /
                          CFLOW=1.0    RFLOW=2E6
                          SHEAT=1000.0   CONDU=6000 /
                          ALPHA=1E-6     TEREF=20.0  /
                          RINFI=0.7     LATEN=6E7    CONSK=4E6  /
                          EXPON=6     ENERG=4000   AFFI0=1E-10  /
                          RCRIT=0.2   FCRIT=15.0   RINTE=0.5   FINTE=100   CUBIC  /
                          TEMAT=20.0    TEMAN=0.5   BETHA=1E-6
   END_SET_#
END_SETS
```

MODEL=41 : J2-Elasto-Visco-Plasticity. Large strain formulation.

The following are the necessary properties to be input to fully define the *J2 Elasto-Visco-Plastic Model* in case of *large strain formulation:* Principal stretches formulation is considered. The model may include isotropic and kinematic hardening.

Parameters:

DENSI=#	Density (compulsory parameter only for dynamic analysis)
YOUNG=#	Elastic modulus.
POISS=#	Poisson's ratio.
YEINI=#	Initial flow stress
YEFIN=#	Saturation flow stress
YEPOW=#	Saturation hardening law exponent
LINHR=#	Linear hardening coefficient
KHARD=#	Linear kinematic hardening
NLKHD=#	Non-linear kinematic hardening
VISCO=#	Viscosity for visco-plastic model
EXPVIS=#	Exponent for plastic viscosity

The isotropic hardening function is defined with the following saturation law (see model 11):

$$\sigma_Y(\zeta) = YEINI + [YEFIN - YEINI]\cdot[1 - \exp(-YEPOW \cdot \zeta)] + LINHR \cdot \zeta$$

where ζ is the equivalent plastic strain.

Example:

```
SETS
   SET=#    <parameters=#>
         [ELEMENT_DATA]     <parameters=#>
         MATERIAL_DATA:   YOUNG=20000.  POISS=0.23  /
                          YEINI=260.  YEFIN=270.  YEPOW=16
   END_SET_#
END_SETS
```

MODEL=42 – 43: Elasticity. Large strain formulation.

The following are the necessary properties to be input to fully define the *Elastic Model* in case of *large strain formulation.*

- Model **42**: Principal stretches formulation
- Model **43**: Neo-Hooke formulation

Parameters:

DENSI=#	Density (compulsory parameter only for dynamic analysis)
YOUNG=#	Elastic modulus.
POISS=#	Poisson's ratio.

Example:

```
SETS
   SET=#    <parameters=#>
         [ELEMENT_DATA]     <parameters=#>
         MATERIAL_DATA:    YOUNG=20000.   POISS=0.3
   END_SET_#
END_SETS
```

MODEL=51 : Thermo-Elasto- Visco-Plastic with or without phase change. Large strain formulation

The following are the necessary properties to be input to fully define the *Thermo-elasto-J2 visco-plastic model with multiphase changes* in case of *large strain formulation.* The model may include both isotropic and kinematic hardening and thermal softening. Most of the properties can be considered as temperature dependent (see below). The model may also reproduce solidification processes with multiphase changes (see below).

Parameters:

DENSI=#	Density (at reference temperature)
YOUNG=#	Elastic modulus
POISS=#	Poisson's ratio
ALPHA=#	Thermal expansion coefficient
SHEAT=#	Specific heat
CONDU=#	Conductivity
YEINI=#	Initial flow stress
YEFIN=#	Saturation flow stress
YEPOW=#	Saturation hardening law exponent
LINHR=#	Linear hardening coefficient
KHARD=#	Linear kinematic hardening
NLKHD=#	Non-linear kinematic hardening
VISCO=#	Viscosity for Norton viscous model (liquidus state) and viscosity for visco-plastic model (solidus state).
EXPVIS=#	Exponent for plastic viscosity

The isotropic hardening function is defined with the following saturation law (see model 11):

$$\sigma_Y(\zeta,\Theta) = YEINI(\Theta) + [YIEFIN(\Theta) - YEINI(\Theta)]\cdot[1 - \exp(-YEPOW\cdot\zeta)] + LINHR(\Theta)\cdot\zeta$$

where Θ is the current absolute temperature and ζ is the equivalent plastic strain.

Change of phase properties:

LATEN=#	Latent heat released.
THOTP=#	Hot phase temperature.
TCOLD=#	Cold phase temperature.
PSTRA=#	Straining due to change of phase (shrinkage)

Temperature dependent properties:

The following properties can be defined as temperature dependent following the procedure explained above.

YOUNG	Elastic modulus
ALPHA	Thermal expansion coefficient
SHEAT	Specific heat
CONDU	Conductivity
YEINI	Initial flow stress
YEFIN	Saturation flow stress
YEPOW	Saturation hardening law exponent
LINHR	Linear hardening coefficient
KHARD	Linear kinematic hardening
NLKHD	Non-linear kinematic hardening
VISCO	Viscosity table
EXPVI	Exponent for plastic viscosity
FRSOL	Solid-fraction (only for phase change analysis)

Example:

```
SETS
   SET=#     <parameters=#>
      [ELEMENT_DATA]          <parameters=#>
      MATERIAL_DATA:        DENSITY=7800.   POISS=0.23    SHEAT=100.0  /
                            ALPHA=1.E-5     VISCO=100.0E6
                            YEINI=260.  YEFIN=270.   YEPOW=16  LINHR=100.  /
                            KHARD=150
      PHASE_CHANGE
         CHANGE=1   LATEN=3.0E5   THOTP=660.0   TCOLD=550.0   PSTRA=0.02
      END_PHASE_CHANGE
      TDEPENDENT_DATA
         CONDU, PIECE_WISE
            20.0   1.5
            700.0  2.1
         END_CONDU
         YOUNG, SPLINE
            20.0   20000
            350.0  22000
            700.0  25000
         END_YOUNG
      END_TDEPENDENT_DATA
   END_SET_#
END_SETS
```

MODEL=52 : Thermo-elasticity with or without phase change. Large strain formulation

The following are the necessary properties to be input to fully define the *Thermo-elasticity model with multiphase changes* in case of *large strain formulation*. Most of the properties can be considered as temperature dependent (see below). The model may also reproduce solidification processes with multiphase changes (see below).

Parameters:

DENSI=#	Density (at reference temperature)
YOUNG=#	Elastic modulus
POISS=#	Poisson's ratio
ALPHA=#	Thermal expansion coefficient
SHEAT=#	Specific heat
CONDU=#	Conductivity
VISCO=#	Viscosity for Norton viscous model (liquidus state).

Change of phase properties:

LATEN=#	Latent heat released.
THOTP=#	Hot phase temperature.
TCOLD=#	Cold phase temperature.
PSTRA=#	Straining due to change of phase (shrinkage)

Temperature dependent properties:

The following properties can be defined as temperature dependent following the procedure explained above.

YOUNG	Elastic modulus
ALPHA	Thermal expansion coefficient
SHEAT	Specific heat
CONDU	Conductivity
VISCO	Viscosity table
FRSOL	Solid-fraction (only for phase change analysis)

Example:

```
SETS
   SET=#    <parameters=#>
      [ELEMENT_DATA]          <parameters=#>
      MATERIAL_DATA:       DENSITY=7800.   POISS=0.23    SHEAT=100.0  /
                           ALPHA=1.E-5     VISCO=100.0E6
      PHASE_CHANGE
         CHANGE=1    LATEN=3.0E5    THOTP=660.0    TCOLD=550.0   PSTRA=0.02
      END_PHASE_CHANGE
      TDEPENDENT_DATA
         CONDU, PIECE_WISE
            20.0    1.5
            700.0  2.1
         END_CONDU
         YOUNG, SPLINE
            20.0    20000
            350.0  22000
            700.0  25000
         END_YOUNG
      END_TDEPENDENT_DATA
   END_SET_#
END_SETS
```

MODEL=81: Mechanical contact equation.

The following are the necessary properties to be input to fully define the *penalty algorithm* for the contact face-to-face MECHANICAL joint-element.

Parameters:

PENN=# Normal penalty parameter.

Example:

```
SETS
   SET=#    <parameters=#>
         [ELEMENT_DATA]      <parameters=#>
         MATERIAL_DATA:    PENN=1.0E+12
   END_SET_#
END_SETS
```

MODEL=82: Thermal contact equation.

The following are the necessary properties to be input to fully define *conduction* model for the contact face-to-face THERMAL joint-element.

Parameters:

HTCCD=# Heat transfer coefficient for conduction model.

Temperature dependent properties:

The following properties can be defined as temperature dependent following the procedure explained above.

HTCCD=# Heat transfer coefficient for conduction model.

Example:

```
SETS
   SET=#    <parameters=#>
         [ELEMENT_DATA]     <parameters=#>
         MATERIAL_DATA:   HTCCD=1000
   END_SET_#
END_SETS
```

MODEL=83: Thermo-Mechanical contact equation.

The following are the necessary properties to be input to fully define the contact face-to-face THERMO-MECHANICAL joint-element constitutive equation. The *penalty algorithm* is assumed for the mechanical partition. Conduction/convection model is assumed for the thermal one.

Parameters:

PENN=#	Normal penalty parameter.
HTCCD=#	Heat transfer coefficient for conduction model.
HTCCV=#	Heat transfer coefficient for convection model.

Temperature dependent properties:

The following properties can be defined as temperature dependent following the procedure explained above.

HTCCD=#	Heat transfer coefficient for conduction model.
HTCCV=#	Heat transfer coefficient for convection model.

Example:

```
SETS
   SET=#    <parameters=#>
         [ELEMENT_DATA]      <parameters=#>
         MATERIAL_DATA:    PENN=1.0E+12, HTCCD=1000, HTCCV=100
   END_SET_#
END_SETS
```

4.3 CONTACT_DATA

compulsory cards if CONTACT_DATA has been input in CONTROL_DATA

Description:

This command cards indicate the beginning of the contact data block.

Structure:

The structure to define the contact data block depends on the formulation chosen:

- ***Small-displacement formulation:***

```
CONTACT_DATA
   <qualifiers>  <parameters=#>
END_CONTACT_DATA
```

It must be defined one line for each slideline to be generated. The automatic generation of contact will be performed duplicating the nodes at the interface between the master set and the slave set.

Parameters:

SLIDE=#	Slideline number
MASTER=#	Master set
SLAVE=#	Slave set
SET=#	Contact set associated to the slideline where is defined the constitutive model.

Qualifiers:

ALL	This qualifier can substitute parameters SLIDE, MASTER, SLAVE if contact generation is required among all materials defined.

Examples:

```
CONTACT_DATA
   SLIDE=1  MASTER=1  SLAVE=2   SET=3
END_CONTACT_DATA
```

```
CONTACT_DATA
   ALL   SET=3
END_CONTACT_DATA
```

- ***Large-displacement formulation:***

```
CONTACT_DATA
    CONTROL_DATA
         SLIDELINE=#   <qualifiers>  <parameters=#>
    END_CONTROL_DATA
    SLIDELINES_DATA
       SLIDELINE=#
          PROPERTIES:   <qualifiers>  <parameters=#>
          SLAVE_NODES
             <slave_surface_topology>
          END_SLAVE_NODES
          MASTER_NODES
             <master_surface_topology>
          END_MASTER_NODES
       END_SLIDELINE_#
    END_SLIDELINES_DATA
END_CONTACT_DATA
```

4.3.1 CONTROL_DATA

Description

This card activates the input procedure for the slidelines control data.

The slidelines control data is made up of one dataset:
(1) SLIDELINE control data where 1 line is input for each slideline;
The structure of the dataset is detailed below.

Structure:

```
CONTROL_DATA
    SLIDELINE=#   <qualifiers>  <parameters=#>
END_CONTROL_DATA
```

SLIDELINE=# <qualifiers> <parameters=#>

compulsory line, one line for each slideline

Parameters:

NSN=#	Number of slave nodes (2D/AXIAL).
NMN=#	Number of master nodes (2D/AXIAL).
NSF=#	Number of slave facets (3D).
NMF=#	Number of master facets (3D).
NNSF=#	Number of nodes per slave facet.
NNMF=#	Number of nodes per master facet.
THMOD=#	Contact model.

Qualifiers:

FRICTIONAL	Coulomb frictional slideline
ENVIRONMENT	Thermal environment slideline.
TWO_PASS	Two_pass algorithm.
SINGLE	Single surface slideline.
GAUSSIAN	Gaussian quadrature rule.
CONVE	Convected norm in the frictional Coulomb law.
PENALTY	Penalty algorithm.
AUGMI	Augmented-Lagrangian algorithm with simultaneous iterations (AUGMENTATION data card must be specified in STRATEGY_DATA block)
AUGMN	Augmented-Lagrangian algorithm with nested loops (AUGMENTATION data card must be specified in STRATEGY_DATA block).
AUGMS	Symmetrized augmented-Lagrangian algorithm with nested loops (AUGMENTATION data card must be specified in STRATEGY_DATA block)

Examples:

```
CONTROL_DATA
        SLIDELINE=1   NSN=10   NMN=29   NNSF=2   NNMF=2  THMOD=3
        SLIDELINE=2   NSN=15   NMN=1    THMOD=4   ENVIRONMENT
END_CONTROL_DATA
```

```
CONTROL_DATA
        SLIDELINE=1   AUGMS    NSN=10   NMN=29
END_CONTROL_DATA
```

4.3.2 SLIDELINE_DATA

Description:

This card activates the input procedure for the slidelines_data.

The slidelines_data is made up of one data block for each slideline.

Each data block is made up of 3 dataset:
(1) slideline PROPERTIES data,
(2) slideline SLAVE_NODES surface topology,
(3) slideline MASTER_NODES surface topology,
The structure of these datasets is detailed below.

Structure:

```
SLIDELINES_DATA
   SLIDELINE=#
      PROPERTIES:   <qualifiers>  <parameters=#>
      SLAVE_NODES
         <slave_surface_topology>
      END_SLAVE_NODES
      MASTER_NODES
         <master_surface_topology>
      END_MASTER_NODES
   END_SLIDELINE_#
SLIDELINES_DATA
```

PROPERTIES **<parameters=#>**

Description:

This card activates the procedure for the input of mechanical/thermal properties of the slideline according to the THMOD defined in the CONTROL_DATA.

THMOD=0 : Mechanical slideline

The following are the necessary properties to be input to fully define the *Mechanical contact for the considered slideline*. This is the default model.

Properties:

PENN=#	Normal penalty
	(Not necessary if either AUTMATIC or INITIAL algorithms are used)
PENT=#	Tangential penalty
	(Not necessary if either AUTMATIC or INITIAL algorithms are used)
FRICT=#	Constant frictional coefficient
FRIC0=#	Constant frictional coefficient (polynomial law)
FRIC1=#	Frictional coefficient (First order term)
FRIC2=#	Frictional coefficient (Second order term)
FRIC3=#	Frictional coefficient (Third order term)
FRIC4=#	Frictional coefficient (Fourth order term)
FRIC5=#	Frictional coefficient (Fifth order term)

The Coulomb frictional coefficient can be assumed as constant parameter:

$$\mu = FRICT$$

or it could be defined according to the following polynomial expression:

$$\mu = FRIC0 + FRIC1 \cdot \alpha + FRIC2 \cdot \alpha^2 + FRIC3 \cdot \alpha^3 + FRIC4 \cdot \alpha^4 + FRIC5 \cdot \alpha^5$$

where α is the frictional dissipation.

Example:

```
SLIDELINE=1
    PROPERTIES:         PENN=10.0E4
    SLAVE_NODES
        <slave_surface_topology>
    END_SLAVE_NODES
    MASTER_NODES
        <slave_surface_topology>
    END_MASTER_NODES
END_SLIDELINE_1
```

```
SLIDELINE=1
    PROPERTIES:         PENN=10.0E4    PENT=10.0E4    FRICT=0.1
    SLAVE_NODES
        <slave_surface_topology>
    END_SLAVE_NODES
    MASTER_NODES
        <slave_surface_topology>
    END_MASTER_NODES
END_SLIDELINE_1
```

THMOD=1 : Mechanical contact + Thermal conduction model

The following are the necessary properties to be input to fully define the *Thermo-Mechanical* contact for the considered slideline: thermal contact is considered specifying a *conduction* heat transfer coefficient.

Properties:

PENN=#	Normal penalty
	(Not necessary if either AUTMATIC or INITIAL algorithms are used)
PENT=#	Tangential penalty
	(Not necessary if either AUTMATIC or INITIAL algorithms are used)
FRICT=#	Constant frictional coefficient
FRIC0=#	Constant frictional coefficient (polynomial law)
FRIC1=#	Frictional coefficient (First order term)
FRIC2=#	Frictional coefficient (Second order term)
FRIC3=#	Frictional coefficient (Third order term)
FRIC4=#	Frictional coefficient (Fourth order term)
FRIC5=#	Frictional coefficient (Fifth order term)
MTHCO=#	Mean slideline thermal conductivity.
GTHCO=#	Gas thermal conductivity.
VICKE=#	Vickers hardness
ROUGH=#	Surface roughness
ASLOP=#	Mean absolute asperity slope
HVCO1=#	Hardness variation coefficient
HVCO2=#	Hardness variation coefficient
COGAS=#	Constitutive gas constant

The Coulomb frictional coefficient can be assumed as constant parameter:

$$\mu = FRICT$$

or it could be defined according to the following polynomial expression:

$$\mu = FRIC0 + FRIC1 \cdot \alpha + FRIC2 \cdot \alpha^2 + FRIC3 \cdot \alpha^3 + FRIC4 \cdot \alpha^4 + FRIC5 \cdot \alpha^5$$

where α is the frictional dissipation.

The heat transfer coeff. for *conduction* can be expressed: $H\left(\mathbf{t}_{\mathrm{n}},\Theta_{g}\right)=H_{S}\left(\mathbf{t}_{\mathrm{n}}\right)+H_{G}\left(\mathbf{t}_{\mathrm{n}},\Theta_{g}\right)$

$$H_{S}\left(\mathbf{t}_{\mathrm{n}}\right)=\frac{1.25\cdot MTHCO\cdot ASLOP}{ROUGH}\left[\frac{\mathbf{t}_{\mathrm{n}}}{\mathrm{HVCO1}}\left(1.62\frac{10^{6}\,ROUGH}{ASLOP}\right)^{-HVCO2}\right]^{\frac{0.95}{1+0.71\cdot HVCO2}}$$

$$H_{G}\left(\mathbf{t}_{\mathrm{n}},\Theta_{g}\right)=\frac{GTHCO}{1.36\cdot ROUGH\cdot\sqrt{-\log\left(5.59\frac{\mathbf{t}_{\mathrm{n}}}{VICKE}\right)}+COGAS\cdot\Theta_{g}}$$

where $\mathbf{t}_{\mathrm{n}}$ is the contact pressure and Θ_{g} is the mean gas temperature.

Example:

```
SLIDELINE=1
    PROPERTIES:  PENN=10.0E4   PENT=10.0E4   FRICT=0.1   /
                 MTHCO= ...   GTHCO= ...    VICKE= ...    ROUGH=...   /
                 ASLOP= ...    HVCO1= ...   HVCO2=...    COGAS= ...
    SLAVE_NODES
        <slave_surface_topology>
    END_SLAVE_NODES
    MASTER_NODES
        <slave_surface_topology>
    END_MASTER_NODES
END_SLIDELINE_1
```

THMOD=2 : Mechanical contact + Thermal conduction model

The following are the necessary properties to be input to fully define the *Thermo-Mechanical* contact for the considered slideline: thermal contact is considered specifying a *conduction* heat transfer coefficient.

Properties:

PENN=#	Normal penalty
	(Not necessary if either AUTMATIC or INITIAL algorithms are used)
PENT=#	Tangential penalty
	(Not necessary if either AUTMATIC or INITIAL algorithms are used)
FRICT=#	Constant frictional coefficient
FRIC0=#	Constant frictional coefficient (polynomial law)
FRIC1=#	Frictional coefficient (First order term)
FRIC2=#	Frictional coefficient (Second order term)
FRIC3=#	Frictional coefficient (Third order term)
FRIC4=#	Frictional coefficient (Fourth order term)
FRIC5=#	Frictional coefficient (Fifth order term)
VICKE=#	Vickers hardness
THCOR=#	Thermal contact resistance
THEXP=#	Exponent

The Coulomb frictional coefficient can be assumed as constant parameter:

$$\mu = FRICT$$

or it could be defined according to the following polynomial expression:

$$\mu = FRIC0 + FRIC1 \cdot \alpha + FRIC2 \cdot \alpha^2 + FRIC3 \cdot \alpha^3 + FRIC4 \cdot \alpha^4 + FRIC5 \cdot \alpha^5$$

where α is the frictional dissipation.

The heat transfer coefficient can be expressed: $$H(\mathbf{t}_n) = THCOR\left(\frac{\mathbf{t}_n}{\text{VICKE}}\right)^{THEXP}$$

where $\mathbf{t}_n$ is the contact pressure.

Example:

```
SLIDELINE=1
    PROPERTIES:   PENN=10.0E4      PENT=10.0E4      FRIC0=0.1    /
                  VICKE=932.   THCOR=1500.   THEXP=0.95
    SLAVE_NODES
        <slave_surface_topology>
    END_SLAVE_NODES
    MASTER_NODES
        <slave_surface_topology>
    END_MASTER_NODES
END_SLIDELINE_1
```

THMOD=3 : Mechanical contact + Thermal conduction and convection model

The following are the necessary properties to be input to fully define the *Thermo-Mechanical* contact for the considered slideline.: thermal contact is considered specifying a *conduction* & *convection* heat transfer coefficients.

Properties:

PENN=#	Normal penalty
	(Not necessary if either AUTMATIC or INITIAL algorithms are used)
PENT=#	Tangential penalty
	(Not necessary if either AUTMATIC or INITIAL algorithms are used)
FRICT=#	Constant frictional coefficient
FRIC0=#	Constant frictional coefficient (polynomial law)
FRIC1=#	Frictional coefficient (First order term)
FRIC2=#	Frictional coefficient (Second order term)
FRIC3=#	Frictional coefficient (Third order term)
FRIC4=#	Frictional coefficient (Fourth order term)
FRIC5=#	Frictional coefficient (Fifth order term)
VICKE=#	Vickers hardness
THCOR=#	Thermal contact resistance
THEXP=#	Exponent

The Coulomb frictional coefficient can be assumed as constant parameter:

$$\mu = FRICT$$

or it could be defined according to the following polynomial expression:

$$\mu = FRIC0 + FRIC1 \cdot \alpha + FRIC2 \cdot \alpha^2 + FRIC3 \cdot \alpha^3 + FRIC4 \cdot \alpha^4 + FRIC5 \cdot \alpha^5$$

where α is the frictional dissipation.

The *conduction* heat transfer coefficient can be expressed:

$$H(\mathbf{t}_n) = THCOR\left(\frac{\mathbf{t}_n}{\text{VICKE}}\right)^{THEXP}$$

where $\mathbf{t}_n$ is the contact pressure. If THEXP=0.0 the conduction heat transfer coefficient is assumed with the following law:

$$H = THCOR$$

Additionally it is necessary to input a mechanical-gap vs. *convection* heat-transfer table with the following structure:

```
GAP_HEAT
    gap_1    heat_1
    gap_2    heat_2
END_GAP_HEAT
```

Example:

```
SLIDELINE=1
    PROPERTIES:   PENN=10.    PENT=10.    FRIC0=0.1    /
                  VICKE=900.0    THCOR=1000    THEXP=1.0
    GAP_HEAT
        0.0          1000
        1.0          500
    END_GAP_HEAT
    SLAVE_NODES
        <slave_surface_topology>
    END_SLAVE_NODES
    MASTER_NODES
        <slave_surface_topology>
    END_MASTER_NODES
END_SLIDELINE_1
```

THMOD=4 : Thermal convection model

The following are the necessary properties to be input to fully define the *Thermal / Thermo-mechanical contact* for the considered slideline: this model permit the simulation of heat transfer due to *convection* between two bodies or *convection with the environment*.

Properties:

ENVTE=# Environment temperature (only if ENVIRONMENT slideline)

It is necessary to input a *Temperature* vs. *Convection-heat-transfer* table with the following structure:

```
TEMP_HEAT
    temp_1  heat_1
    temp_2  heat_2
END_TEMP_HEAT
```

To be noted that the value of the heat transfer coefficient at each node of the slideline will be obtained coming into this table with the temperature of the SLAVE nodes. The SLAVE temperature drives the heat flux through the slideline.

Note:

In case of convection with the environment it is necessary to declare ENVIRONMENT qualifier in the SLIDELINE data in the CONTROL_DATA card, is not necessary to define the number of master nodes NMN or facets NMF and the MASTER_NODE data block definition must be omitted.

Example:

```
SLIDELINE=1
    PROPERTIES:  ENVTE=20.0
    TEMP_HEAT
        300.0       100.0
        500.0        50.0
    END_TEMP_HEAT
    SLAVE_NODES
        <slave_surface_topology>
    END_SLAVE_NODES
END_SLIDELINE_1
```

```
SLIDELINE=2
    PROPERTIES:   NONE
    TEMP_HEAT
         300.0     1.0
        1500.0     0.5
    END_TEMP_HEAT
    SLAVE_NODES
        <slave_surface_topology>
    END_SLAVE_NODES
    MASTER_NODES
        <slave_surface_topology>
    END_MASTER_NODES
END_SLIDELINE_2
```

THMOD=5 : Mechanical contact + Thermal conduction and convection model

The following are the necessary properties to be input to fully define the *Thermo-Mechanical* contact for the considered slideline.: thermal contact is considered specifying a *conduction* & *convection* heat transfer coefficients.

Properties:

PENN=#	Normal penalty
	(Not necessary if either AUTMATIC or INITIAL algorithms are used)
PENT=#	Tangential penalty
	(Not necessary if either AUTMATIC or INITIAL algorithms are used)
FRICT=#	Constant frictional coefficient
FRIC0=#	Constant frictional coefficient (polynomial law)
FRIC1=#	Frictional coefficient (First order term)
FRIC2=#	Frictional coefficient (Second order term)
FRIC3=#	Frictional coefficient (Third order term)
FRIC4=#	Frictional coefficient (Fourth order term)
FRIC5=#	Frictional coefficient (Fifth order term)
VICKE=#	Vickers hardness
THCOR=#	Thermal contact resistance
THEXP=#	Exponent
AIRCO=#	Gas conductivity used for convection model.

The Coulomb frictional coefficient can be assumed as constant parameter:

$$\mu = FRICT$$

or it could be defined according to the following polynomial expression:

$$\mu = FRIC0 + FRIC1 \cdot \alpha + FRIC2 \cdot \alpha^2 + FRIC3 \cdot \alpha^3 + FRIC4 \cdot \alpha^4 + FRIC5 \cdot \alpha^5$$

where α is the frictional dissipation.

The *conduction* heat transfer coefficient can be expressed:

$$H(\mathbf{t}_n) = THCOR\left(\frac{\mathbf{t}_n}{\text{VICKE}}\right)^{THEXP}$$

where $\mathbf{t}_n$ is the contact pressure. If THEXP=0.0 the conduction heat transfer coefficient is assumed with the following law:

$$H = THCOR$$

Convection heat-transfer is computed according to the mechanical-gap using the following expression:

$$H = \min\left(THCOR; \frac{AIRCO}{MechGap}\right)$$

Example:

```
SLIDELINE=1
    PROPERTIES:   PENN=10.   /
                  THCOR=1000.   AIRCO=100.
    SLAVE_NODES
        <slave_surface_topology>
    END_SLAVE_NODES
    MASTER_NODES
        <slave_surface_topology>
    END_MASTER_NODES
END_SLIDELINE_1
```

SLAVE_NODES

Description:

This card activates the procedure for the input of the slave surface topology for the current slideline.

Structure:

- (2D/AXIAL) Input one record for each nodal point that define the slave surface. The total number of records must be equal to NSN defined in the CONTROL_DATA.

IPOIN	Nodal point number (local numeration)
SLAVE_NODE	Nodal point number (global numeration)

- (3D) Input one record for each facet that define the slave surface. The total number of records must be equal to NSF defined in the CONTROL_DATA.

IFACET	Facet number
NODE_1	First nodal point number that define de facet (global numeration)
NODE_#	#th nodal point number that define de facet (global numeration)
NODE_NNSF	Last nodal point number that define de facet according to the NNSF defined in the CONTROL_DATA (global numeration)

Note:

The numeration of each edge-facet must be input according to the outer normal rule. (See examples).

Examples:

See MASTER_NODES section

MASTER_NODES

Description:

This card activates the procedure for the input of the master surface topology for the current slideline.

Structure:

- (2D/AXIAL) Input one record for each nodal point that define the master surface. The total number of records must be equal to NMN defined in the CONTROL_DATA.

IPOIN	Nodal point number (local numeration)
MASTER_NODE	Nodal point number (global numeration)

- (3D) Input one record for each facet that define the master surface. The total number of records must be equal to NMF defined in the CONTROL_DATA.

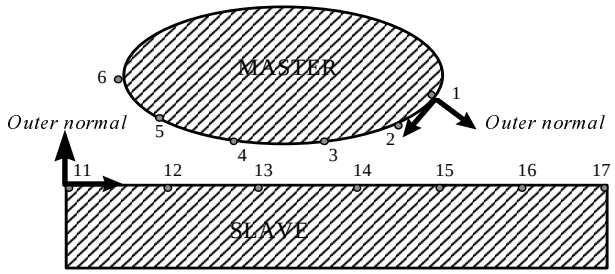
IFACET	Facet number
NODE_1	First nodal point number that define de facet (global numeration)
NODE_#	#th nodal point number that define de facet (global numeration)
NODE_NNMF	Last nodal point number that define de facet according to the NNMF defined in the CONTROL_DATA (global numeration)

Note:

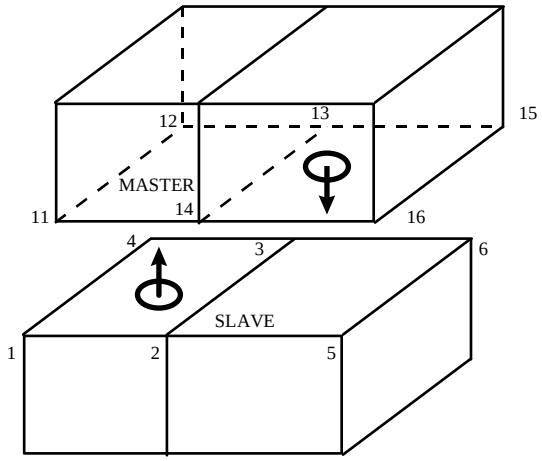
The numeration of each edge-facet must be input according to the outer normal rule. (See examples).

Examples:

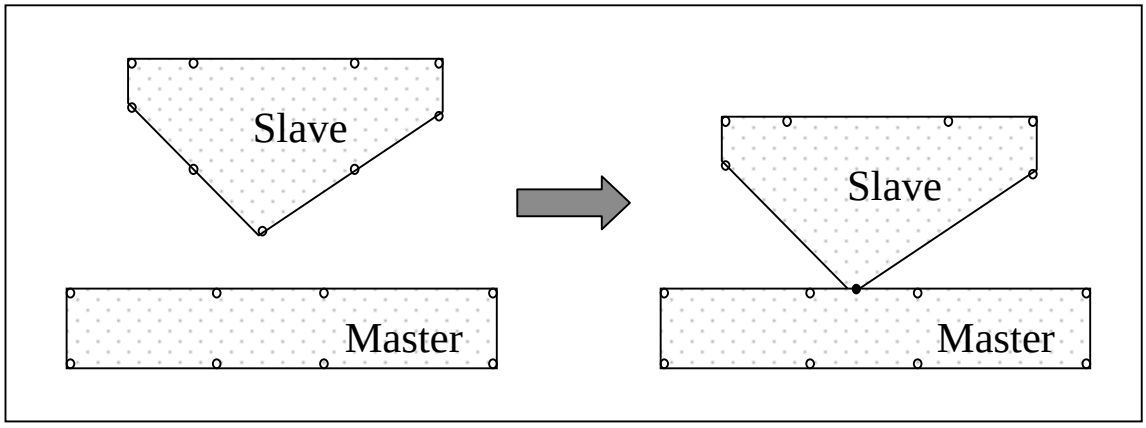
```
SLIDELINES_DATA
   SLIDELINE=#
      PROPERTIES:   <qualifiers>  <parameters=#>
      SLAVE_NODES                 ! (2D/AXIAL)
         1,  11
         2,  12
         3,  13
         4,  14
         5,  15
         6,  16
         7,  17
      END_SLAVE_NODES
      MASTER_NODES
         1,  1
         2,  2
         3,  3
         4,  4
         5,  5
         6,  6
      END_MASTER_NODES
   END_SLIDELINE_#
SLIDELINES_DATA
```



```
SLIDELINES_DATA
   SLIDELINE=#
      PROPERTIES:   <qualifiers>  <parameters=#>
      SLAVE_NODES                 ! (3D)
         1,  1,2,3,4
         2,  3,2,5,6
      END_SLAVE_NODES
      MASTER_NODES
         1,  11,12,13,14
         2,  14,13,15,16
      END_MASTER_NODES
   END_SLIDELINE_#
SLIDELINES_DATA
```

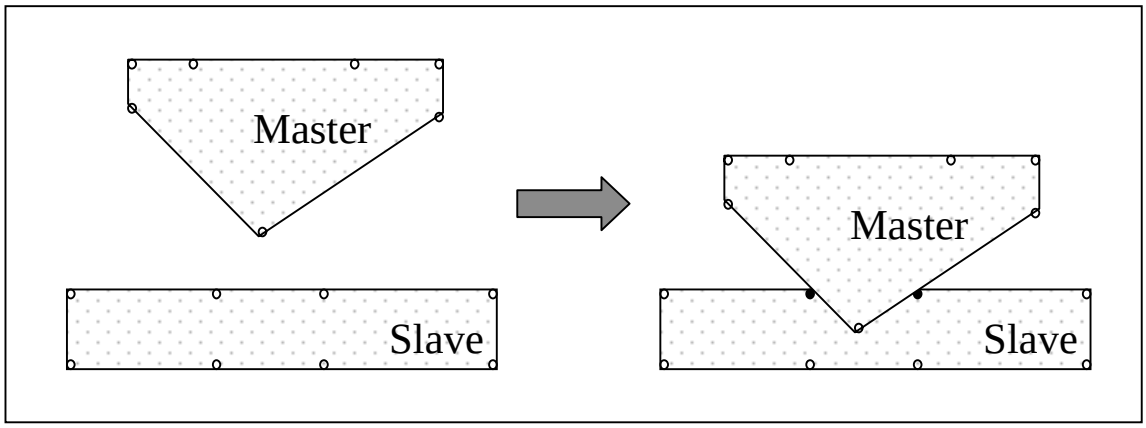


Rules of slave and master in case of mechanical analysis:

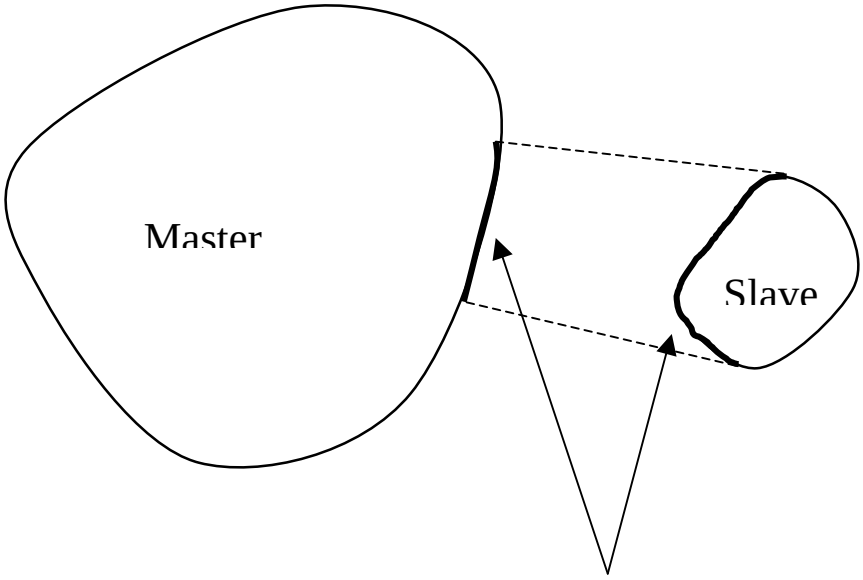


Any slave node cannot penetrate the master surface: there is no check on the master nodes.

If option two_pass is used both slave and master nodes penetration is checked.

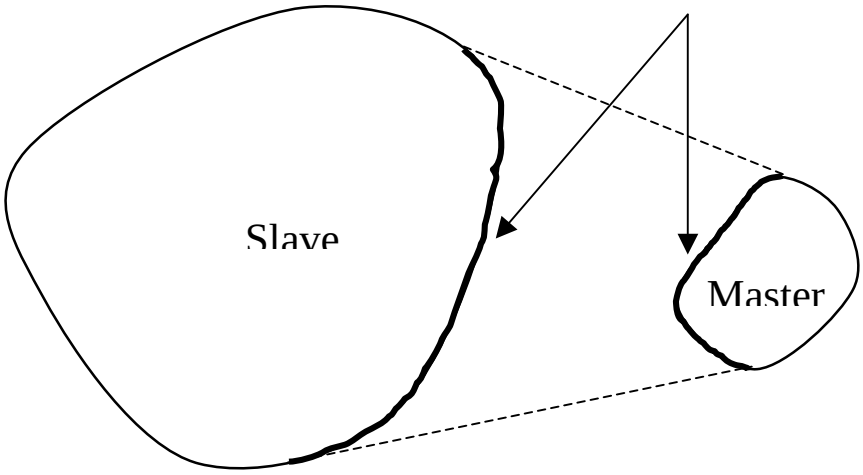


Rules of slave and master in case of heat convection analysis:



Heat transfer due to heat convection between the slave and master surfaces.

The slave nodes are projected on the master surface so that depending on the rules (slave-master) of the surfaces the results could be different.



5. INTERVAL_DATA COMMANDS

The INTERVAL_DATA commands are used to specify the beginning and end of the data corresponding to a time interval. They also specify the number and size of the time steps to be used in the interval. Interval data are supplied according to prescribed formats.

INTERVAL_DATA <parameters=#>

END_INTERVAL_DATA

compulsory cards

Description:

This command cards indicate the beginning and the end of each interval data block.

Parameters:

NSTEP=# Number of steps for this time interval;

DTIME=# Initial time step increment of this interval.

DTMIN=# Minimum time step increment permitted within this interval (only if automatic LOAD_INCREMENTATION is active).

DTMAX=# Maximum time step increment permitted within this interval (only if automatic LOAD_INCREMENTATION is active).

Notes:

When automatic load incrementation is used, the parameter DTIME is computed according to the LOAD_INCREMENTATION strategy card and it will be defined in the range that varies between DTMIN and DTMAX.

Structure:

```
INTERVAL_DATA
    FUNCTION <qualifiers>
        <functions definition>
    END_FUNCTION
    LOAD <qualifiers>
        <load conditions>
    END_LOAD
    BOUNDARY <qualifiers>
        <boundary conditions>
    END_BOUNDARY
    INITIAL_DATA
        <initial_data options>
    END_INITIAL_DATA
    ACTIVATION
        <activation options>
    END_ACTIVATION
    STRATEGY <qualifiers>
        <strategy options>
    END_STRATEGY
END_INTERVAL_DATA
```

5.1 FUNCTION *<qualifiers>*

compulsory card

Description:

This command activates the input procedure for the functions. These functions determine one factor used in loads, boundaries or any time dependent item for each time interval. The curves type and its relative parameters must be input immediately after this card with the format specified here.

Qualifiers :

NEW_FUNCTION	It specifies that a new functions will be defined;
OLD_FUNCTION	It specifies to keep the same functions of the previous time interval.

Notes:

- In case of NEW_FUNCTION this card must be followed by the function data block cards and by the END_FUNCTION card (refer to Function Data Option Card).
- In case of OLD_FUNCTION it is not necessary to write the END_FUNCTION card.
- In a START RUN for the first interval it is required that the qualifier specifies NEW_FUNCTION.

Structure:

Each curve is made up of 1 record:
(1) FUNCTION definition,
The structure of these record is detailed below.

This card must be repeated for as many functions as needed. The total number of functions must be less or equal MCURV, the maximum number of curves already specified in CONTROL_DATA.

```
FUNCTION  <qualifiers>
    FUNCTION=#     <qualifiers>   <parameters=#>
END_FUNCTION
```

5.1.1 FUNCTION *<qualifiers> <parameters=#>*

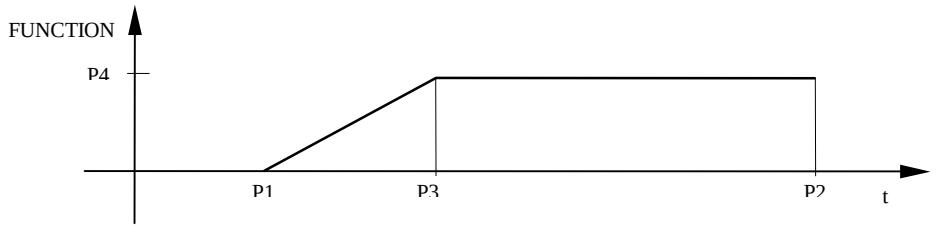
optional card

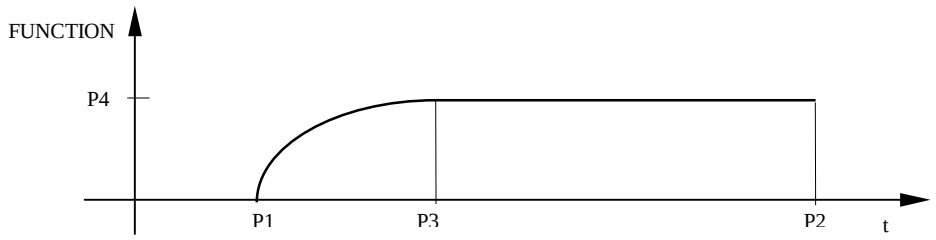
Qualifiers:

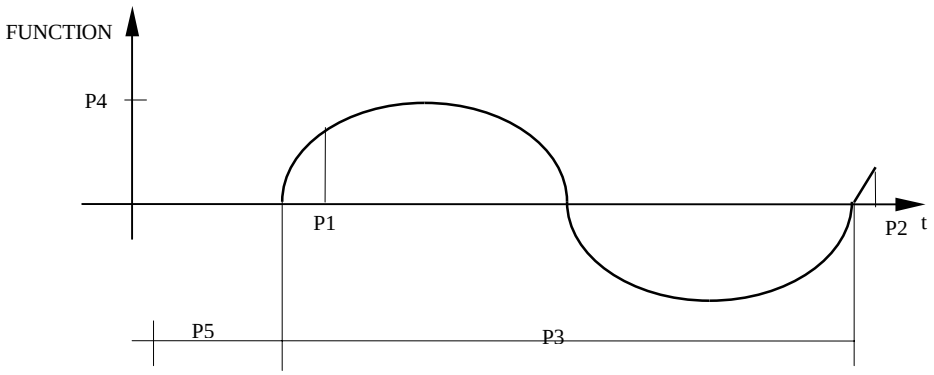
LINEAR	Linear rising/constant,
PARABOLIC	Parabolic rising/constant,
SINE	Sinusoidal signal.
SAW_TOOTH	Saw-tooth signal.
DISCRETE	To be input by points: the first point must be (0.0, 0.0) (see examples)

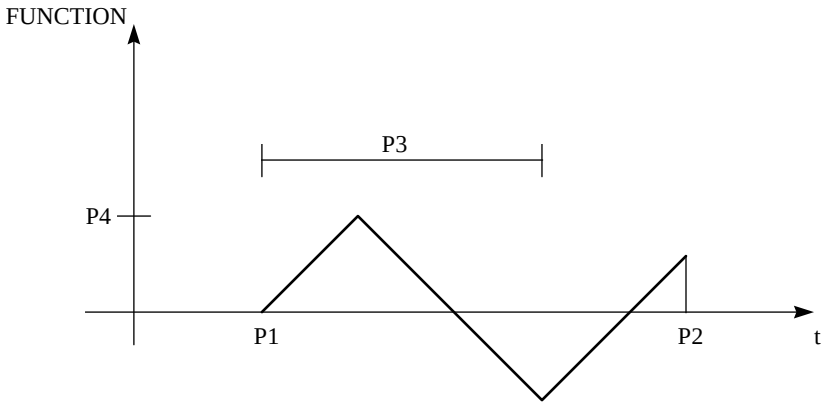
Parameters:

START=P1	Starting time
END=P2	Finishing time
TIME=P3	For load types LINEAR & PARABOLIC: rising time
	For load types SINE & SAW_TOOTH: period time
AMPLITUDE=P4	Maximum amplitude.









Examples:

```
FUNCTION:   NEW_FUNCTION
FUNCTION=1: PARABOLIC, start=10,  end=20, time=10, amplitude=2
FUNCTION=2: LINEAR, start=20, end=40, time=30, amplitude=1
END_FUNCTION
```

```
FUNCTION:   NEW_FUNCTION
    FUNCTION=3   DISCRETE
        0.0   0.0       !  time_1   value_1
        1.0   10.0      !  time_2   value_2
        2.0   15.0      !  time_3   value_3
        3.0   5.0       !  time_4   value_4
    END_DISCRETE
END_FUNCTION
```

5.2 LOCAL_SYSTEM *<qualifiers>*

optional card

Description:

This command activates the input procedure for the local axes systems to be used to prescribe boundary conditions or loads in a different system of the global one. The rotation matrix ROT (U_global=ROT*U_local) must be input to define the new system.

Qualifiers :

NEW_LOCAL	It specifies that a new local axes system will be defined;
OLD_LOCAL	It specifies to keep the same local axes system of the previous time interval.

Notes:

- In case of NEW_LOCAL this card must be followed by the local data block cards and by the END_LOCAL.
- In case of OLD_LOCAL it is not necessary to write the END_LOCAL card.
- In a START RUN for the first interval it is required that the qualifier specifies NEW_LOCAL.

Structure:

Each local system is made up of 1 record that must be repeated for as many local system as needed. The structure is the following

```
LOCAL_SYSTEM <qualifiers>
    LOCAL=#        <parameters=#>
END_LOCAL_SYSTEM
```

Parameters:

ROT11=# ROT12=#	
ROT21=# ROT22=#	Rotation matrix coefficients for 2D/AXIALSYMM problem
ROT11=# ROT12=# ROT13=#	
ROT21=# ROT22=# ROT23=#	
ROT32=# ROT31=# ROT33=#	Rotation matrix coefficients for 3D problem

Example:

```
LOCAL_SYSTEM:   NEW_LOCAL
LOCAL=1:        ROT11=1.0  ROT12=0.0  /
                ROT21=0.0  ROT22=1.0
END_LOCAL_SYSTEM
```

```
LOCAL_SYSTEM:   NEW_LOCAL
LOCAL=1:        ROT11=1.0  ROT12=0.0  ROT13=0.0  /
                ROT21=0.0  ROT22=1.0  ROT23=0.0  /
                ROT32=0.0  ROT31=0.0  ROT33=5.0
END_LOCAL_SYSTEM
```

5.3 LOAD *<qualifiers>*

compulsory card

Description:

Gravity, point, surface and volume loads are defined. The time functions governing the application of the loads are also assigned.

Qualifiers :

NEW_LOAD	It specifies that a new load and/or boundary conditions must be applied;
KEEP	It specifies that the reactions obtained at the end of the previous interval will taken into account.
OLD_LOAD	It specifies to keep the same load and boundary conditions as in the previous time interval.

Notes:

- In case of NEW_LOAD this card must be followed by the load data block cards and by the END_LOAD card (refer to LOAD_DATA Option Card).
- In case of OLD_LOAD it is not necessary to write the END_LOAD card.
- In a START RUN for the first interval it is required that the qualifier specifies NEW_LOAD.
- This block is a compulsory block even if some problems may not need the specification of load conditions: in this case write simply:

```
LOAD: NEW_LOAD
END_LOAD
```

Structure:

The load_data block is made up of the following records:

(1) TITLE definition (optional),
(2) FUNCTION definition (compulsory),
(3) GRAVITY_LOAD definition (optional),
(4) POINT_LOAD definition (optional),
(5) FACE_LOAD definition (optional),
(6) FACE_FLUX definition (optional),
(7) SOURCE_LOAD definition (optional),
(8) PRINT_LOAD card (optional)

The structure of these records is detailed below.

```
LOAD: NEW_LOAD
    TITLE:        <title>
    FUNCTION=#          <qualifiers>
    GRAVITY_LOAD:    <parameters>
    POINT_LOAD
        <qualifiers>   <point_load_parameters=#>
    END_POINT_LOAD
    FACE_LOAD
        <qualifiers>   <face_load_parameters=#>
    END_FACE_LOAD
    FACE_FLUX
        <qualifiers>   <face_flux_parameters=#>
    END_FACE_FLUX
    SOURCE_LOAD
        <qualifiers>   <sources_parameters=#>
    END_SOURCE_LOAD
    PRINT_LOAD
END_LOAD
```

```
LOAD: OLD_LOAD
```

```
LOAD: NEW_LOAD
END_LOAD
```

5.3.1 TITLE: *<title>*

optional card

Description:

This card specifies the existence of a load title. In this case, the command must be followed by the relative title card which have at most 74 characters.

Examples:

```
LOAD: NEW_LOAD
    TITLE:  gravity load and point load
    FUNCTION=#          <qualifiers>
    GRAVITY_LOAD:    <parameters>
    POINT_LOAD
        <qualifiers>   <point_load_parameters=#>
    END_POINT_LOAD
    FACE_LOAD
        <qualifiers>   <face_load_parameters=#>
    END_FACE_LOAD
    FACE_FLUX
        <qualifiers>   <face_flux_parameters=#>
    END_FACE_FLUX
    SOURCE_LOAD
        <qualifiers>   <sources_parameters=#>
    END_SOURCE_LOAD
    PRINT_LOAD
END_LOAD
```

5.3.2 FUNCTION=# *<qualifier>*

optional card

Description:

This card selects a time function to control the application of the interval loads. The function value at each time is multiplied by the reference load to obtain the effective load.

Qualifiers :

ABSOLUTE	It specifies that the function must be evaluated with the total time.
RELATIVE	It specifies that the function must be evaluated with the interval time.

Default:

FUNCTION = 0 ! Always return zero as factor.

Examples:

```
LOAD: NEW_LOAD
    TITLE:        <title>
    FUNCTION=1    ABSOLUTE
    GRAVITY_LOAD:    <parameters>
    POINT_LOAD
        <qualifiers>   <point_load_parameters=#>
    END_POINT_LOAD
    FACE_LOAD
        <qualifiers>   <face_load_parameters=#>
    END_FACE_LOAD
    FACE_FLUX
        <qualifiers>   <face_flux_parameters=#>
    END_FACE_FLUX
    SOURCE_LOAD
        <qualifiers>   <sources_parameters=#>
    END_SOURCE_LOAD
    PRINT_LOAD
END_LOAD
```

5.3.3 GRAVITY_LOAD *<parameters>*

optional card

Description:

Only for MECHANICAL and COUPLED problems. It is necessary to define the gravity direction, input component by component in the global system and the value of the gravity acceleration.

Parameters:

GX=#	unit vector component in x direction
GY=#	unit vector component in y direction
GZ=#	unit vector component in z direction
GRAVY=#	value of the gravity acceleration

Notes:

The program actually accumulates all contributions of the point, surface and gravity loads in a reference loads which is then multiplied by a load factor determined on the base of the specified load/displacement functions. Thus, since the gravity load usually does not vary with time, it is necessary to define a first interval in which only the gravity load may act. After the first interval, all the remaining forces may be considered.

Examples:

```
LOAD: NEW_LOAD
    TITLE:        <title>
    FUNCTION=#          <qualifiers>
    GRAVITY_LOAD:   Gx=0.0, Gy=0.0, Gz=-1.0,  GRAVY=9.8
    POINT_LOAD
        <qualifiers>   <point_load_parameters=#>
    END_POINT_LOAD
    FACE_LOAD
        <qualifiers>   <face_load_parameters=#>
    END_FACE_LOAD
    FACE_FLUX
        <qualifiers>   <face_flux_parameters=#>
    END_FACE_FLUX
    SOURCE_LOAD
        <qualifiers>   <sources_parameters=#>
    END_SOURCE_LOAD
    PRINT_LOAD
END_LOAD
```

5.3.4 POINT_LOAD

optional card

Description:

This card specifies the existence of external point loads. In this case, this card must be followed by the relative load data in the format reported as follows.

This load has to be considered as a reference load since the actual value for each time step is obtained by multiplying this reference load by a factor determined from the contribution of all defined load functions.

Structure:

Card Set 1: Applied load cards

NODE	Node number.
FX=# FY=# FZ=# FT=#	Value of the pressure at node (n) in X, Y and Z direction

Notes:

- For axialsymmetric problems, the point loads input should be the total loading on the circumferential ring passing through the nodal point concerned.
- For THERMAL problems FT is the thermal load

Examples:

```
LOAD: NEW_LOAD
   TITLE:        <title>
   FUNCTION=#          <qualifiers>
   GRAVITY_LOAD:    <parameters>
   POINT_LOAD
      NODE=100   FX=1000.0   FY=0.0
      NODE=101   FX=500.0    FY=0.0
   END_POINT_LOAD
   FACE_LOAD
      <qualifiers>   <face_load_parameters=#>
   END_FACE_LOAD
   FACE_FLUX
      <qualifiers>   <face_flux_parameters=#>
   END_FACE_FLUX
   SOURCE_LOAD
      <qualifiers>   <sources_parameters=#>
   END_SOURCE_LOAD
   PRINT_LOAD
END_LOAD
```

5.3.5 FACE_LOAD

optional card

Description:

Only for MECHANICAL and COUPLED problems. This card specifies the existence of external distributed surface loads. In this case, this card must be immediately followed by the relative load data which structure is reported in the following. Enter as many loaded faces as required.
This load has to be considered as a reference load since the actual value for each time step is obtained by multiplying this reference load by a factor determined from the contribution of all defined load functions.

Structure:

Card set 1: Distributed edge/face load card

ELEM=#	The element with which the element edge/face is associated.
NODES=#,#,…	List of the nodal points, in sequence, forming the element edge/face on which the distributed load acts.

Card Set 2: Distributed edge/face load cards

GLOBAL	Global components definition
FX=# FY=# FZ=#	Value of the pressure at node in X, Y and Z directions
	(One card for each node of the edge/face)
LOCAL	Local components definition
FX=Ft=0.0 FY=Fn=# FZ=Fb=0.0	Value of the pressure at node in Tangent, Normal and Bi-normal directions. Fn>0 means in the direction defined by the outer normal
	(One card for each node of the edge/face)

Notes :

- Subsets 1, and 2 must be repeated in turn for every element edge/face on which a distributed load acts. The element edges/faces can be considered in any order.
- The numbering of the nodes in the edge/face for quadratic elements must start with the corner nodes, followed by the midside node. (See GEOMETRY in GENERAL_DATA)

Examples:

```
LOAD: NEW_LOAD
   TITLE:        <title>
   FUNCTION=#          <qualifiers>
   GRAVITY_LOAD:    <parameters>
   POINT_LOAD
      <qualifiers>  <point_load_parameters=#>
   END_POINT_LOAD
   FACE_LOAD
      ELEM=10   NODES=1,2,3,4
      GLOBAL     FX=10.0  FY=150.0  FZ=0.0  /
                 FX=10.0  FY=150.0  FZ=1.0  /
                 FX=10.0  FY=150.0  FZ=2.0  /
                 FX=10.0  FY=150.0  FZ=0.0
      ELEM=11   NODES=5,6,7,8
      LOCAL      FX=0.0   FY=1000.0  FZ=0.0  /
                 FX=0.0   FY=1000.0  FZ=0.0  /
                 FX=0.0   FY=1000.0  FZ=0.0  /
                 FX=0.0   FY=1000.0  FZ=0.0
   END_FACE_LOAD
   FACE_FLUX
      <qualifiers>  <face_flux_parameters=#>
   END_FACE_FLUX
   SOURCE_LOAD
      <qualifiers>  <sources_parameters=#>
   END_SOURCE_LOAD
   PRINT_LOAD
END_LOAD
```

5.3.6 FACE_FLUX

optional card

Description:

Only for THERMAL and COUPLED problems. This card specifies the existence of external distributed surface fluxes. In this case, this card must be immediately followed by the relative load data which structure is reported in the following. Enter as many loaded faces as required.
This load has to be considered as a reference load since the actual value for each time step is obtained by multiplying this reference load by a factor determined from the contribution of all defined load functions.

Structure:

Card set 1: Distributed edge/face flux card

ELEM=#	The element with which the element edge/face is associated.
NODES=#,#,…	List of the nodal points, in sequence, forming the element edge/face on which the distributed load acts.

Card Set 2: Distributed edge/face load cards

FT_1=# FT_2=# FT_n=# Value of the heat flux at (One card for each node of the edge/face)

Notes :

- Subsets 1, and 2 must be repeated in turn for every element edge/face on which a distributed load acts. The element edges/faces can be considered in any order.
- The numbering of the nodes in the edge/face for quadratic elements must start with the corner nodes, followed by the midside node. (See GEOMETRY in GENERAL_DATA)

Examples:

```
LOAD: NEW_LOAD
    TITLE:        <title>
    FUNCTION=#          <qualifiers>
    GRAVITY_LOAD:    <parameters>
    POINT_LOAD
        <qualifiers>   <point_load_parameters=#>
    END_POINT_LOAD
    FACE_LOAD
        <qualifiers>   <face_load_parameters=#>
    END_FACE_LOAD
    FACE_FLUX
        ELEM=10    NODES=1,2,3,4
                    FT_1=10.0   FT_2=10.0   FT_3=0.   FT_4=10.0
    END_FACE_FLUX
    SOURCE_LOAD
        <qualifiers>   <sources_parameters=#>
    END_SOURCE_LOAD
    PRINT_LOAD
END_LOAD
```

5.3.7 SOURCE_LOAD

optional card

Description:

This card specifies the existence of thermal sources loads. In this case, this card must be followed by the relative load data in the format reported as follows.

This load has to be considered as a reference load since the actual value for each time step is obtained by multiplying this reference load by a factor determined from the contribution of all defined load functions.

Structure:

Card set 1: Volume load card

ELEM=#	The element to which the load is associated
SOURCE=#	The value of the thermal volume load

Examples:

```
LOAD: NEW_LOAD
    TITLE:        <title>
    FUNCTION=#          <qualifiers>
    GRAVITY_LOAD:    <parameters>
    POINT_LOAD
        <qualifiers>   <point_load_parameters=#>
    END_POINT_LOAD
    FACE_LOAD
        <qualifiers>   <face_load_parameters=#>
    END_FACE_LOAD
    FACE_FLUX
        <qualifiers>   <face_flux_parameters=#>
    END_FACE_FLUX
    SOURCE_LOAD
        ELEM=200    SOURCE=1.0
        ELEM=201    SOURCE=10.0
        ELEM=202    SOURCE=11.0
    END_SOURCE_LOAD
    PRINT_LOAD
END_LOAD
```

5.3.8 PRINT_LOAD

optional card

Description:

This card specifies the printing of the calculated equivalent nodal components of the reference load obtained by summing up all type of loads (point, distributed and body forces) acting in this time interval.

Examples:

```
LOAD: NEW_LOAD
    TITLE:        <title>
    FUNCTION=#          <qualifiers>
    GRAVITY_LOAD:    <parameters>
    POINT_LOAD
        <qualifiers>  <point_load_parameters=#>
    END_POINT_LOAD
    FACE_LOAD
        <qualifiers>  <face_load_parameters=#>
    END_FACE_LOAD
    FACE_FLUX
        <qualifiers>  <face_flux_parameters=#>
    END_FACE_FLUX
    SOURCE_LOAD
        <qualifiers>  <sources_parameters=#>
    END_SOURCE_LOAD
    PRINT_LOAD
END_LOAD
```

5.4 BOUNDARY *<qualifier>*

compulsory card

Description:

This card activates the input procedures for the Dirichlet boundary conditions including prescribed displacements or temperatures. Immediately after this card the boundary conditions must be input with a format as described here.

Qualifiers:

NEW_BOUNDARY	It specifies that a new boundary conditions is required;
KEEP	It specifies that the reactions obtained at the end of the previous interval will be taken into account.
OLD_BOUNDARY	It specifies that the old boundary conditions are maintained.

Notes:

- The prescribed value of the restrained nodal degrees of freedom has to be considered as a reference value. In fact the actual value for each time step is obtained by multiplying this reference value by the factor determined from the specified time interval.
- In case of OLD_BOUNDARY it is not necessary to write the END_BOUNDARY card.
- In a START RUN for the first interval it is required that the qualifier specifies NEW_BOUNDARY.
- This block is a compulsory block even if some problems may not need the specification of boundaty conditions (for instance, a THERMAL problem): in this case write simply:

```
BOUNDARY: NEW_BOUNDARY
END_BOUNDARY
```

Structure:

The boundary_data block is made up of the following records:

(1) FUNCTION definition (compulsory),

(2) BOUNDARY_RECORDS definition (compulsory),

The structure of these records is detailed below.

```
BOUNDARY: NEW_BOUNDARY
   FUNCTION=#          <qualifiers>
   [BOUNDARY_RECORD]     <nodes_list>,  <restrictions>,  <values>
END_BOUNDARY
```

```
BOUNDARY: OLD_BOUNDARY
```

```
BOUNDARY: NEW_BOUNDARY
END_BOUNDARY
```

5.4.1 FUNCTION=# *<qualifier>*

If any function is specified, this card must appear immediately after the BOUNDARY card

Description:

This card selects one function as factor. The function value at each time is multiplied by the reference prescribed displacements to obtain the effective prescribed displacements.

Qualifiers :

ABSOLUTE	It specifies that the function must be evaluated with the total time.
RELATIVE	It specifies that the function must be evaluated with the interval time.

Default:

FUNCTION = 0 ! Always return zero as factor.

Examples:

```
BOUNDARY: NEW_BOUNDARY
   FUNCTION=1   ABSOLUTE
   [BOUNDARY_RECORD]      <nodes_list>,  <restrictions>,  <values>
END_BOUNDARY
```

5.4.2 [BOUNDARY_RECORD] *<nodes_list>, <restrictions>, <values>*

NODE_LIST	List of restrained nodes with identical restriction (max. 56 nodes).
IFIX_X, IFIX_Y, IFIX_Z, IFIX_T	Flags for fixity of restrained nodes: Each digit is: 0 - not prescribed d.o.f. 1 - prescribed d.o.f.
VAL_X, VAL_Y, VAL_Z, VAL_T	The reference prescribed value for 1st d.o.f.
LOCAL=#	Local system of boundary conditions defined in LOCAL_DATA block.

Notes :

- The order of the dof is as follows:
    1. X displacement (or temperature for THERMAL problems)
    2. Y displacement
    3. Z displacement (only 3D)
    4. Temperature (for COUPLED problems)
- If the list of restrained nodes is sequential, the following format may be used:

  1 TO 9 SKIP 2 110 0.000 0.000 0.000

  This will apply the stated condition to nodes 1,3,5,7,9.

  The default for SKIP is 1.
- If SET option is used then all the nodes of the specified set will be prescribed. The format is the following:

  SET=1 110 0.000 0.000 0.000

Examples:

```
BOUNDARY: NEW_BOUNDARY
   FUNCTION=1
      1 TO 100          1101    1.0   1.0  0.0   10.0   LOCAL=1
      101, 105, 108     0010    0.0   0.0  3.0   0.0
      119               1110    1.0   10.0 9.0   0.0
END_BOUNDARY
```

```
BOUNDARY: NEW_BOUNDARY
   FUNCTION=1
      SET=1             1110    1.0   10.0 9.0   0.0
END_BOUNDARY
```

5.5 INITIAL_DATA

Description:

This command cards indicate the beginning and the end of the initial data block.

Initial conditions for acceleration, velocity and displacement are defined. Initial data are supplied according to prescribed formats.

Structure:

```
INITIAL_DATA
    DISPLACEMENT
        <initial displacement conditions>
    END_DISPLACEMENT
    VELOCITY
        <initial velocity conditions>
    END_VELOCITY
    ACCELERATION
        <initial acceleration conditions>
    END_ACCELERATION
    TEMPERATURE
        <initial temperature conditions>
    END_TEMPERATURE
END_INITIAL_DATA
```

5.5.1 DISPLACEMENT

optional card

Description:

Only for MECHANICAL or COUPLED problems. This card implies the existence of an initial displacement field that must be read.

Structure:

Input one record for each nodal point with non zero initial displacement:

IPOIN	Nodal point number
DISPL X	Initial Displacement X at node
DISPL Y	Initial Displacement Y at node
DISPL Z	Initial Displacement Z at node

Notes:

- If the number of points is less than the total number of points, then the remaining nodes automatically set to zero displacement.
- If the list of nodes is sequential, the following format may be used:

  1 TO 9 SKIP 2 10.000 0.000 5.000

This will apply the stated initial displacement components to nodes 1,3,5,7,9.

The default for SKIP is 1.

Example:

```
INITIAL_DATA
   DISPLACEMENT
      1 TO 10          1.0   1.5   0.0
      11, 12, 13       1.0   0.0   0.0
      14               1.0   1.0   1.0
   END_DISPLACEMENT
END_INITIAL_DATA
```

5.5.2 VELOCITY

optional card

Description:

Only for MECHANICAL or COUPLED dynamic problems. This card implies the existence of an initial velocity field that must be read.

Structure:

Input one record for each nodal point with non zero initial velocity:

IPOIN	Nodal point number
VELOC X	Initial Velocity X at node
VELOC Y	Initial Velocity Y at node
VELOC Z	Initial Velocity Z at node

Notes:

- If the number of points is less than the total number of points, then the remaining nodes automatically set to zero initial velocity.
- If the list of nodes is sequential, the following format may be used:

  1 TO 9 SKIP 2 10.000 10.000 0.000

This will apply the stated condition to nodes 1,3,5,7,9.

The default for SKIP is 1.

Example:

```
INITIAL_DATA
   VELOCITY
      1 TO 10          1.0  1.5  0.0
      11, 12, 13       1.0  0.0  0.0
      14               1.0  1.0  1.0
   END_VELOCITY
END_INITIAL_DATA
```

5.5.3 ACCELERATION

optional card

Description:

Only for MECHANICAL or COUPLED dynamic problems . This card implies the existence of an initial acceleration field that must be read.

Structure:

Input one record for each nodal point with non zero initial acceleration:

IPOIN	Nodal point number
ACCEL X	Initial Acceleration X at node
ACCEL Y	Initial Acceleration Y at node
ACCEL Z	Initial Acceleration Z at node

Notes:

- If the number of points is less than the total number of points, then the remaining nodes automatically set to zero acceleration.
- If the list of nodes is sequential, the following format may be used:

  1 TO 9 SKIP 2 1.000 0.000 0.000

This will apply the stated initial acceleration components to nodes 1,3,5,7,9.

The default for SKIP is 1.

Example:

```
INITIAL_DATA
   ACCELERATION
      1 TO 10          1.0  1.5  0.0
      11, 12, 13       1.0  0.0  0.0
      14               1.0  1.0  1.0
   END_ACCELERATION
END_INITIAL_DATA
```

5.5.4 TEMPERATURE

optional card

Description:

Only for THERMAL or COUPLED problems. This card implies the existence of an initial temperature field that must be read.

Structure:

Input one record for each nodal point with non zero temperature:

IPOIN	Nodal point number.
SET	Set number.
TEMPE	Initial temperature at node.

Notes:

- If the number of points is less than the total number of points, then the remaining nodes automatically set to zero absolute temperature.
- If the list of nodes is sequential, the following format may be used:

  1 TO 9 SKIP 2 273.000

This will apply the stated condition to nodes 1,3,5,7,9.

The default for SKIP is 1.

- If SET option is used then the specified initial temperature is fixed for all the nodes of the selected set.

Example:

```
INITIAL_DATA
   TEMPERATURE
      SET=1          20.0
   END_TEMPERATURE
END_INITIAL_DATA
```

```
INITIAL_DATA
   TEMPERATURE
      1 TO 10        27.0
      11, 12, 13     1000.0
      14             400.0
   END_TEMPERATURE
END_INITIAL_DATA
```

5.6 ACTIVATION *<qualifiers>*

optional card

Description:

This command is used to define the input procedure for the activation/deactivation procedure. Using this command it is possible to activate/deactivate a certain set of element or/and contact slideline in the specified interval.

Structure:

The structure of this data block is detailed below. It is possible to observe that it is possible to activate/deactivate ELEMETS, SETS, or SLIDELINES.

ACTIVATION *<qualifiers>*		
[ELEMENT_RECORD]	<elements_list>	*<qualifiers>*
[SET_RECORD]	<sets_list>	*<qualifiers>*
[SLIDELINE_RECORD]	<slidelines_list>	*<qualifiers>*
END_ACTIVATION		

5.6.1 [ELEMENT_RECORD] *<elements_list> <qualifiers>*

Description:

This card is used to activate/deactivate ELEMENTS during the analysis.

Elements_list :

ELEMENT=#	It specifies a generical list of elements to activate/deactivate.
#1 TO #n	It specifies a sequential list of elements to be activated/deactivated. It is also possible to use the command SKIP=# to skip elements in a sequential list with the defined increment. The default for SKIP is 1.

Qualifiers :

ON	It specifies that the element_list must be activated.
OFF	It specifies that the element_list must be deactivated but its database is saved (it is possible to acticate it again in another interval) .
DELETE	It specifies that the element_list must be deactivated but its database is removed (it is NO possible to acticate it again in another interval) .

Examples:

```
ACTIVATION
        ELEMENT= 1        TO 100     ON
        ELEMENT= 101      TO 200     OFF
        ELEMENT= 201 , 205           DELETE
        ELEMENT= 301 TO 100     SKIP=2   ON
END_ACTIVATION
```

5.6.2 [SET_RECORD] *<set_list> <qualifiers>*

Description:

This card is used to activate/deactivate SETS during the analysis.

Set_list :

ELEMENT=#	It specifies a generical list of sets to activate/deactivate.
#1 TO #n	It specifies a sequential list of sets to be activated/deactivated. It is also possible to use the command SKIP=# to skip sets in a sequential list with the defined increment. The default for SKIP is 1.

Qualifiers :

ON	It specifies that the set_list must be activated.
OFF	It specifies that the set_list must be deactivated but its database is saved (it is possible to acticate it again in another interval) .
DELETE	It specifies that the set_list must be deactivated but its database is removed (it is NO possible to acticate it again in another interval) .

Examples:

```
ACTIVATION
        SET= 1       TO 10    ON
        SET= 11    TO 20      OFF
        SET= 21                DELETE
END_ACTIVATION
```

5.6.3 [SLIDELINE_RECORD] *<slideline_list> <qualifiers>*

Description:

This card is used to activate/deactivate SLIDELINES during the analysis.

Slidelines_list :

ELEMENT=#	It specifies a generical list of slidelines to activate/deactivate.
#n1 TO #nn	It specifies a sequential list of slidelines to be activated/deactivated. It is also possible to use the command SKIP=# to skip slidelines in a sequential list with the defined increment. The default for SKIP is 1.

Qualifiers :

ON	It specifies that the slideline_list must be activated.
OFF	It specifies that the slideline_list must be deactivated but its database is saved (it is possible to acticate it again in another interval) .
DELETE	It specifies that the slideline_list must be deactivated but its database is removed (it is NO possible to acticate it again in another interval) .

Examples:

```
ACTIVATION
      SLIDELINE= 1      TO 10    ON
      SLIDELINE= 11    TO 20     OFF
      SLIDELINE= 21               DELETE
END_ACTIVATION
```

5.7 STRATEGY *<qualifier>*

compulsory card

Description:

In this section the commands concerning the solution strategy adopted for the time interval is defined.

Qualifiers :

NEW_STRATEGY	It specifies that a new solution strategy must be followed;
OLD_STRATEGY	It specifies to keep the same solution strategy as in the previous time interval.

Notes:

- In case of NEW_STRATEGY this card must be followed by the strategy data block cards and by the END_STRATEGY card (refer to Strategy Data Option Card).
- In case of OLD_STRATEGY it is not necessary to write the END_STRATEGY card.
- In a START RUN for the first interval it is required that the qualifier specifies NEW_STRATEGY.

Structure:

The strategy_data block is made up of the following records:

(1) ACCELERATION definition (optional),
(2) ALGORITHM definition (compulsory),
(3) ARC_LENGTH definition (optional),
(4) AUGMENTATION definition (optional),
(5) CONVERGENCE definition (compulsory),
(6) DYNAMIC definition,
(7) LINE_SEARH definition (optional),
(8) LOAD_INCREMENTATION definition (optional),
(9) OUTPUT definition (optional),
(10) PLOT definition (optional),
(11) POST_PROCESS definition (optional),
(12) SAVE definition (optional),
(13) SOLVER definition (optional),
(14) TERMINATION definition (optional),
(15) TIME_INTEGRATION definition (optional),
(16) TIME_STEPPING definition (optional),

The structure of these records is detailed below.

STRATEGY: NEW_STRATEGY	
ACCELERATION:	*<qualifiers>*
ALGORITHM:	*<qualifiers> <parameters =#>*
ARC_LENGTH:	*<qualifiers> <parameters =#>*
AUGMENTATION:	*<parameters =#>*
CONVERGENCE:	*<qualifiers> <parameters =#>*
DYNAMIC:	*<qualifiers>*
LINE_SEARCH:	*<qualifiers>*
LOAD_INCREMENTATION:	*<qualifiers> <parameters =#>*
OUTPUT:	*<qualifiers> <parameters =#>*
PLOT:	*<qualifiers> <parameters =#>*
POST_PROCESS:	*<qualifiers> <parameters =#>*
SAVE:	*<qualifiers> <parameters =#>*
SOLVER:	*<qualifiers> <parameters =#>*
TERMINATION:	*<parameters =#>*
TIME_INCREMENTATION:	*<qualifiers> <parameters =#>*
TIME_STEPPING:	*<qualifiers> <parameters =#>*
END_STRATEGY	

STRATEGY: OLD_STRATEGY

5.7.1 ACCELERATION *<qualifier>*

optional card

Description:

This card indicates if it is required to activate or to turn off an convergence acceleration methods. If these algorithms are activated:

- the first trial solution at each time step is determined on the basis of the last converged result.
- subsequent iterative displacements are modified according to the selected algorithm.

Qualifiers:

OFF	Convergence accelerator is set to 'OFF'
ON	Only the first trial solution at each time step is accelerated
SN1	One parameter Secant-Newton
SN2	Two parameters Secant-Newton
BFGS	BFGS Quasi-Newton

Default:

OFF Unless specified differently in the previous time interval

Incompatibilities:

ARC_LENGTH if status different from OFF

Examples:

```
STRATEGY: NEW_STRATEGY
    …….
    ACCELERATION:  BFSG
END_STRATEGY
```

5.7.2 ALGORITHM *<qualifier> <parameter=#>*

optional card

Description:

This card specifies the frequency of updating the stiffness matrix during the current time interval.

Qualifiers:

INITIAL	Do not update the initial stiffness matrix
UPDATE	Update the stiffness matrix with a frequency as specified (Newton- Raphson algorithms)
FULL	Full Newton-Raphson (update at each iteration and time step)

Parameters:

STEP=#	Modified Newton-Raphson #>0 Update until time step > # #<0 Update every $\|\#\|$ time steps
ITERATION=#	Modified Newton-Raphson #>0 Update until iteration > # #<0 Update every $\|\#\|$ iterations

Default:

INITIAL	Unless specified differently in the previous time interval

Examples:

```
STRATEGY: NEW_STRATEGY
   …….
   ALGORITHM:  UPDATE   FULL                ! Update every iteration of each step
END_STRATEGY
```

```
STRATEGY: NEW_STRATEGY
   …….
   ALGORITHM:  UPDATE   STEP=-5              ! Update every five steps
END_STRATEGY
```

5.7.3 ARC_LENGTH *<qualifier> <parameters=#>*

to be considered only for MECHANICAL STATIC or THERMAL STEADY_STATE analysis

Description:

This card indicates if it is required to activate or to turn off the arch-length algorithms. The qualifier then specifies the type of arch-length method have to be used.

Qualifiers:

OFF	Disactivate any arch-length algorithm previously defined;
NORMAL	Normal plane algorithm;
UPDATED	Updated normal plane algorithm;
SPHERICAL	Spherical path algorithm;
DISPLACEMENT	Displacement control algorithm.

Parameters:

LENGTH=#	Prescribed value of the arc-length per time step. For DISPLACEMENT control only:
NODE=#	Node number of the controlling degree of freedom;
DOF=#	Controlling degree of freedom.

Default:

NO_ARC_LENGTH	Unless specified differently in the previous time interval

Incompatibilities:

ACCELERATION, SMOOTHING, TIME_STEPPING, TIME_INTEGRATION

Notes:

ARC_LENGTH requires the use of automatic load incrementation unless LENGTH parameter is used

Examples:

```
STRATEGY: NEW_STRATEGY
    …….
    ARC_LENGTH:   SPHERICAL  LENGTH=1.0
END_STRATEGY
```

```
STRATEGY: NEW_STRATEGY
    …….
    ARC_LENGTH:   DISPLACEMENT   NODE=23    DOF=1   LENGTH=1.0
END_STRATEGY
```

5.7.4 AUGMENTATION *<parameters=#>*

compulsory card in case of augmented lagrangian contact algorithms

Description:

This card indicates if it is required to activate the augmentation loop in case of augmented lagrangian contact algorithms.

Parameters:

GTOLE=#	Convergence tolerance based on normal gap (%)
PTOLE=#	Convergence tolerance based on normal traction or pressure (%)
MITER#	Maximum number of augmentations permitted

Examples:

```
STRATEGY: NEW_STRATEGY
    …….
    AUGMENTATION:   GTOLE=1.0     MITER=5
END_STRATEGY
```

5.7.5 CONVERGENCE *<qualifiers> <parameters=#>*

compulsory card

Description:

This card indicates the type of convergence check to be adopted.

Qualifiers:

TOTAL	Convergence is checked comparing the norm of the residual forces with the norm of the total external forces;
INCREMENTAL	Convergence is checked comparing the norm of the residual forces with the norm of the incremental external forces;
DISPLACEMENT or TEMEPERATURE	Convergence is checked comparing the norm of the iterative displacements (or temperatures) with the norm of the incremental displacements (or temperatures);
ENERGY	Convergence is checked comparing the energy dissipated by the iterative residual forces with the energy dissipated by the incremental external forces. Not to be used together with LINE_SEARCH strategy.
SKIP	Skip to next time step after MITER iterations even if not converged

Parameters:

TOLER=#	Convergence tolerance (%)
MITER#	Maximum number of iterations to converge.
TSKIP=#	It is possible to skip to the following time-step according to a skipping tolerance (%) and …..
MSKIP=#	….. to a minimum number of iterations. If SKIP option is actived the following defaults are assumed TSKIP = TOLER and MSKIP = MITER.

Default:

Unless specified differently in the previous time interval

CONVERGENCE: TOTAL TOLER=1.0 MITER=10

Example:

```
STRATEGY: NEW_STRATEGY
   …….
   CONVERGENCE:  DISPLACEMENT   TOLER=1.0   MITER=20
END_STRATEGY
```

5.7.6 DYNAMIC *<qualifiers>*

to be considered only for DYNAMIC or TRANSIENT problems

Description:

This card switches to quasi-static or dynamic conditions.

Qualifiers:

OFF	To specify that the analysis will continue as quasi-static (neglecting time derivatives terms)
ON	To specify that the analysis will continue as dynamic (taking into account time derivative terms)

Default:

DYNAMIC: ON

Examples:

```
STRATEGY: NEW_STRATEGY
    …….
    DYNAMIC: OFF
END_STRATEGY
```

5.7.7 LINE_SEARCH <*qualifier*> <*parameters*=#>

to be considered only for MECHANICAL analysis;

Description:

This card indicates if it is required to activate or to turn off the line search method for accelerating the convergence in a non linear problem.

Qualifiers:

ON	Line search is set to 'ON'
OFF	Line search is set to 'OFF'

Parameters:

STOLR=#	Line search tolerance
LSMIN=#	Minimum line search allowed step
LSMAX=#	Maximum line search allowed step
MLOOP=#	Maximum number of line search trials

Default:

- OFF Unless specified differently in the previous time interval
- The default parameters when LINE_SEARCH status is ON are:
  STOLR=0.3 LSMIN=0.1 LSMAX=2.0 MLOOP=5

Examples:

```
STRATEGY: NEW_STRATEGY
   …….
   LINE_SEARCH:   ON   STOLR=0.3   LSMIN=0.1   LSMAX=2.0  MLOOP=5
END_STRATEGY
```

5.7.8 LOAD_INCREMENTATION *<qualifiers> <parameters=#>*

Optional card.

Description:

This card indicates if its required an automatic load/displacement incrementation. The available options are specified by the qualifiers. However, for the first step the load factor dF1 is specified in the 'interval command card' as initial time increment. The second load factor instead. is computed as the 20% of dF1.

Qualifier:

OFF — No automatic load/displacement incrementation (incremental load factor computed according to loading/displacement functions)

Parameters:

NITER=# — Incremental load factor is computed in order to keep the number of equilibrium iterations constant to a selected number (NITER=number of iterations).

$$dF_i = dF_{i-1} \frac{N^o \text{ desired iterations}}{N^o \text{ performed iterations}}$$

STIF1=# — Incremental load factor is computed in order to keep the change of the stiffness parameter (within the time step) constant to a selected stiffness change dS (%) ; to be considered only for MECHANICAL STATIC analysis; compulsory card when ARCL_LENGTH option is activated

$$dF_i = dF_{i-1} \frac{\text{desired } dS}{\text{actual } (S_{i-1} - S_i)}$$

STIF2=# — Incremental load factor is computed in order to keep the ratio of the change of the stiffness parameter (within the time step) with respect to the stiffness parameter of the previous time step constant to a selected stiffness change ratio dS/S (%); to be considered only for MECHANICAL STATIC analysis; compulsory card when ARCL_LENGTH option is activated

$$dF_i = dF_{i-1} \frac{\text{desired } dS / S}{\text{actual } (S_{i-1} - S_i) / S_{i-1}}$$

ARC_LENGTH=# — Incremental load factor is computed in order to keep the arc length constant to #. To use this option the ARC_LENGTH algorithm must have been activated previously. If # = 0 last arc length is keep constant; to be considered only for MECHANICAL STATIC analysis; compulsory card when ARCL_LENGTH option is activated

COOLR =# — Value of temperature decrement to achieve within the following time-step in a self-driven cooling analysis; It can be used only for THERMAL or THERMO-MECHANICAL analysis.

Default:

LOAD_INCREMENTATION: OFF

Incompatibilities:

DYNAMIC, COUPLED, TIME_STEPPING, TIME_INCREMENTATION

Examples:

```
STRATEGY: NEW_STRATEGY
    …….
    LOAD_INCREMENTATION:    NITER=10
END_STRATEGY
```

```
STRATEGY: NEW_STRATEGY
    …….
    LOAD_INCREMENTATION:    ARC_LENGTH=1.0
END_STRATEGY
```

```
STRATEGY: NEW_STRATEGY
    …….
    LOAD_INCREMENTATION:    COOLR=30.0
END_STRATEGY
```

5.7.9 OUTPUT *<qualifier> <parameters=#>*

optional card

Description:

This command activates the output procedures for printing results of the analysis. It is necessary to define as many cards as many output options are needed.

Qualifiers:

NONE	Do not output at all
ALL	Output everything at requested frequency
REACTIONS	Output Reactions at requested frequency
DISPLACEMENTS	Output Displacements at requested frequency
STRESSES	Output Gaussian Global Stresses at requested frequency
PRINCIPAL	Output Gaussian Principal Stresses at requested freq.
INTERNAL	Output Other Gaussian Variables at requested frequency
NDGLB	Output Nodal Global Stresses at requested frequency
NDPRI	Output Nodal Principal Stresses at requested freq.
CONTACT REACTIONS	Output Nodal contact reactions for each slideline
CONTACT SLIDE	Output Nodal contact information for each slideline
CONTACT ALL	Output Nodal all contact information about each slideline

Parameters:

STEP=#	Step frequency of dumping of converged results
ITERATION=#	Iteration frequency of dumping of results
	#>0 Dump results if converged after # iterations
	#<0 Dump results every $\|\#\|$ iterations

Examples:

```
STRATEGY: NEW_STRATEGY
   …….
   OUTPUT:  DISPLACEMENTS   STEP=5
   OUTPUT:  PRINCIPAL  STEP=1   ITERATION=-3
END_STRATEGY
```

5.7.10 PLOT *<qualifier> <parameters=#>*

optional card

Description:

This command activates the output procedures to the X-Y plot files.

Qualifier:

NEW_PLOT	The following definition is for new curves. Save the old ones.
APPEND	The following definition is appended to the old ones. All curves will continue be plotted.

Structure:

The user may define as many X_Y plots as he wants during the run, with a total maximum of 40. Each curve is defined with the structure:

1. TITLE: <title> (compulsory)
2. X: <qualifiers>, <parameters=#> (X-axis definition)
3. Y: <qualifiers>, <parameters=#> (Y-axis definition)

After the last curve definition the card : END_PLOT must be entered.

Qualifiers:

NSTEP	Number of time step
TIME	Time
LOAD_FACTOR	Load function factor
BOUNDARY_FACTOR	Boundary function factor
NITER	Number of iterations
NITE1	Number of iterations for partition 1
NITE2	Number of iterations for partition 2
STIFF	Current stiffness parameter
FORCE CH	Component CH of Force (reaction) at a node (or difference between two nodes) CH = (X,Y,Z,T)
DISPLACEMENT CH	Component CH of Displacement at a node (or difference between two nodes) CH = (X,Y,Z,T)
VELOCITY CH	Component CH of Velocity at a node (or difference between two nodes) CH = (X,Y,Z,T)
ACCELERATION CH	Component CH of Acceleration at a node (or difference between two nodes) CH = (X,Y,Z,T)

REACTION CH	Component CH of Reaction at a node (or difference between two nodes) CH = (X,Y,Z,T)
STRESS CH	Component CH of Stress at a Gauss Point CH = (XX,YY,XY,ZZ,XZ,YZ,P1,P2,P3)
STRAIN CH	Component CH of Strain at a Gauss Point CH = (XX,YY,XY,ZZ,XZ,YZ,P1,P2,P3)
INTERNAL N	Number N of Internal Variables at a Gauss Point
AUXILIAR N	Number N of Auxiliary Variables at a Gauss Point
TEMPERATURE T	Temperature at node T (or difference between two nodes)

Parameters:

NODE=#	For FORCE/DISPLACEMENT/VELOCITY/ACCELERATION/ TEMPERATURE number of node
ELEMENT=#	For STRESS/STRAIN/INTERNAL/AUXILIAR number of element
NODE1=#	For difference of FORCE/DISPLACEMENT/ VELOCITY/ACCELERATION/ TEMPERATURE number of 1st.
NODE2=#	For difference of FORCE/DISPLACEMENT/ VELOCITY/ACCELERATION/ TEMPERATURE number of 2nd.
INT_POINT=#	For STRESS/STRAIN/INTERNAL/AUXILIAR number of Gauss Point (1-NGAUL)

Default:

NO_PLOT

Example:

```
STRATEGY: NEW_STRATEGY
    …….
    PLOT
        TITLE: curve 1
            X: TIME
            Y: LAMBDA
        TITLE: curve 2
            X:NSTEP
            Y:FORCE X: NODE 10
        TITLE: curve 3
            X:TIME
            Y:DISPLACEMENT Y: NODE1=10, NODE2=11
        TITLE: curve 4
            X:VELOCITY Z: NODE 1
            Y:FORCE Y: NODE=18
        TITLE: curve 3
            X:TIME
            Y:TEMPERATURE T: NODE1=10, NODE2=11
        TITLE: curve 5
            X:TIME
            Y:STRESS XX: ELEMENT 1 INT_POINT 3
        TITLE: curve 6
            X:STRAIN P1: ELEMENT 1, INT_POINT 3
            Y:STRESS P1: ELEMENT 1, INT_POINT 3
        TITLE: curve 7
            X:TIME
            Y:INTERNAL,18,ELEMENT=2,INT_POINT=4
        TITLE: curve 8
            X:NSTEP
            Y:VELOCITY,P,NODE1=2,NODE2=4
    END_PLOT
END_STRATEGY
```

5.7.11 POST_PROCESS *<qualifiers> <parameters=#>*

optional card

Description:

This command activates the output procedures to the post-process files.

Qualifiers:

DISPLACEMENTS	Post-process of the displacements fields.
TEMPERATURES	Post-process of the temperature field.
CONTACT	Post-process of the contact information (only there exists contact).
INTERNAL	Post-process of the internal variables according to the constitutive model.
STRESSES	Post-process of the stress tensor.
J2STR	Post-process of the J2 deviatoric stress indicator
PRESS	Post-process of the pressure (trace of the stress tensor).
ALLSTR	Post-process of the STRESSES, PRESS and J2STR.
TSTRAIN	Post-process of the total strain tensor (elastic + plastic + thermal +….).
J2TOT	Post-process of the J2 deviatoric indicator.
VTOT	Post-process of trace of the strain tensor.
ALLTOT	Post-process of the TSTRAIN, J2TOT and VTOT.
ESTRAIN	Post-process of the elastic strain tensor.
J2ELA	Post-process of the J2 deviatoric indicator.
VELA	Post-process of trace of the elastic strain tensor.
ALLELA	Post-process of the ESTRAIN, J2ELA and VELA.
PSTRAIN	Post-process of the plastic strain tensor.
J2PLA	Post-process of the J2 deviatoric indicator.
VPLA	Post-process of trace of the plastic strain tensor.
ALLPLA	Post-process of the PSTRAIN, J2PLA and VPLA.
NONE	General post-process procedure is deactivated in the current interval.
OFF	The post-process of the indicated variable is deactivated in the current interval.

Parameters:

STEP=#	Step frequency of dumping of converged results

Default:

NONE

Examples:

```
STRATEGY: NEW_STRATEGY
    …….
    POST_PROCESS: NONE
END_STRATEGY
```

```
STRATEGY: NEW_STRATEGY
    …….
    POST_PROCESS: STRESSES    STEP=3
    POST_PROCESS: INTERNAL    OFF
END_STRATEGY
```

5.7.12 SAVE <*qualifier*> <*parameters=#*>

Description:

This card indicates if it is required to activate save procedure to have the possibility to use the restart facilities.

Qualifiers:

NONE	No restart file will be created.
ONLY_ONE	Only the last step in the last interval will be save d in the restart file.
APPEND	It will be saved sequentially in the restart file the last step of each interval specified in the analysis.

Parameters:

STEP=#	The restart file will updated every # steps.

Default:

It will not generate the restart file.

Examples:

```
STRATEGY: NEW_STRATEGY
    …….
    SAVE:  APPEND  STEP=5
END_STRATEGY
```

5.7.13 SOLVER *<qualifiers> <parameters=#>*

optional command

Description:

This command specifies the system equation solver (linear set of equations) to be used in the analysis.

Qualifiers:

DIRECT_SKYLINE	Direct solver using a skyline storage.
• RENUMBER	Internal node renumbering
• NO_RENUMBER	Suppress internal node renumbering.
ITERATIVE	If the partition is SYMMETRIC will be used a CONJUGATE GRADIENT iterative solver. In case of UNSYMMETRIC partition will be used a GMRES iterative solver.
• DIAGONAL	Diagonal matrix pre-conditioning
• LEFT	Left matrix pre-conditioning.
• RIGHT	Right matrix pre-conditioning.
• TOTAL	Global matrix pre-conditioning
SYMMETRIC	Symmetric system of equations for all the partitions
UNSYMMETRIC	Non-symmetric system of equations for all the partitions
UNSYMMETRIC=#	Non-symmetric system of equations for partition number #.

Parameters:

MAXIT=#	Maximum number of iterations admitted for ITERATIVE solvers.
TOLER=#	Convergence TOLERANCE for ITERATIVE solvers based on the norm of the right hand side of the system of equations. Default: TOLER=0.1E-6 .
KRYLO=#	Dimension of the Krylov Subspace for the ITERATIVE UNSYMMMETRIC solver (GMRES). Default: KRYLO=20.

Default:

SOLVER: DIRECT SKYLINE SYMMETRIC RENUMBER

Notes:

- This command can be alternatively defined in the STRATEGY_DATA block.
- The RENUMBER option can be used if it exists a physical connection between all the continuum body involved in the analysis: the connection could be achieved through the definition of contact slidelines.
- For coupled analysis with two partitions (mechanical and thermal) the UNSYMMETRIC qualifier may be specified for a particular partition.

  Example:

UNSYMMETRIC	means both partitions are unsymmetric;
UNSYMMETRIC=1	means only the first partition is unsymmetric;
UNSYMMETRIC=2	means only the second partition is unsymmetric;

- The qualifiers are grouped above to show their exclusivity. If combinations of qualifiers from a group are used the last one will always be utilised. Some qualifiers in certain combinations may be ignored. See the examples below.

Example:

```
CONTROL_DATA
   GEOMETRY       <qualifiers>
   MECHANICAL     <qualifiers>
   DIMENSIONS     <parameters=#>
   CONTACT        <qualifiers>  <parameters=#>
   SOLVER         DIRECT SKYLINE      SYMMETRIC  RENUMBER
   POST_PROCESS   <qualifiers>
   SMOOTHING      <qualifiers>
END_CONTROL_DATA
```

5.7.14 TERMINATION *<parameters=#>*

compulsory card when an automatic LOAD_INCREMENTATION is activated.

Description:

This card indicates if its required an automatic end of the analysis.

Parameters:

END_TIME=#	Specified the time (relative to the INTERVAL) to be reached to stop the analysis (only for THERMAL analysis).
MIN_TEMPE=#	Specified the minimum temperature to be reached in every nodes of the mesh to stop the analysis (only for THERMAL analysis).
COOLR=#	Specified the minimum cooling rate to be reached in every nodes of the mesh to stop the analysis (only for THERMAL analysis).
STIFI=#	Specified final stiffness parameter. (in percentage). The current stiffness parameter is defined as the ratio between the norm of the incremental applied forces and the work done by the incremental forces themselves on the incremental displacement in the last time step. Thus this parameter tends to zero as the incremental displacements tend to larger value for the same applied incremental forces. (only for QUASI-STATIC MECHANICAL analysis).

.

Examples:

```
STRATEGY: NEW_STRATEGY
    …….
    TERMINATION:    END_TIME=120
END_STRATEGY
```

```
STRATEGY: NEW_STRATEGY
    …….
    FINAL_STIFFNESS:    STIFI=10.0 [%]
END_STRATEGY
```

5.7.15 TIME_INCREMENTATION <*qualifier*> <*parameter=#*>

to be considered only for THERMAL TRANSIENT analysis with TIME_STEPPING WILSON

Description:

This card specifies the time marching scheme to be adopted during the current time interval.

Qualifiers:

CONSTANT	Time increment constant and constant theta parameter
LOG1	Time increment constant in log scale and constant theta parameter

Parameter:

INCREM=#	Number of subintervals in each log-interval

Default:

TIME_INCREMENTATION: CONSTANT

Incompatibilities:

LOAD_INCREMENTATION

Notes:

The update of Theta parameter concerns only the Theta-Wilson time stepping algorithm.

Examples:

```
STRATEGY: NEW_STRATEGY
    …….
    TIME_INCREMENTATION: LOG1  INCREM=10
END_STRATEGY
```

5.7.16 TIME_STEPPING *<qualifier> <parameters=#>*

to be considered only for DYNAMIC analysis;

Description:

This card specifies the time-stepping integration scheme to be adopted during the current time interval.

Qualifiers:

NEWMARK	Newmark's time-stepping algorithm (2nd order)

Parameters:

BETA=#	BETA for Newmark's method (> 0.5). Default value $\beta = 0.5$
GAMMA=#	GAMMA for Newmark's method. Default value $\gamma = 0.25$.
	If no value is input, $\gamma = (\beta + 0.5)^2 / 4$ to ensure unconditional stability.

Incompatibilities:

ACCELERATION, ARC_LENGTH, LOAD_INCREMENTATION

Examples:

```
STRATEGY: NEW_STRATEGY
   …….
   TIME_STEPPING:  NEWMARK   BETA=0.5  GAMMA=0.25
END_STRATEGY
```